

UNIVERSITÀ DI PISA

Facoltà di Ingegneria  
Corso di Dottorato in Ingegneria Aerospaziale  
*XX Ciclo*

Tesi di Dottorato

**Studio di metodologie integrate per la determinazione  
dei dati aria nei velivoli Fly-by-Wire**



**Relatori**

Prof. Eugenio DENTI  
Prof. Roberto GALATOLO

**Candidata**

Veronica POGGI

**Direttore del Corso di Dottorato**

Prof.ssa Maria Vittoria SALVETTI

18 LUGLIO 2008

*A me, perchè ora inizia il bello!*

*E il celo fici occhio.*

Andrea Camilleri

## Sommario

La presente tesi di Dottorato affronta lo sviluppo di metodologie per l'integrazione dei sensori dati aria nei sistemi di controllo di tipo *Fly-by-Wire*, (FBW).

La determinazione dei dati aria ovvero della pressione statica, della pressione totale e degli angoli d'incidenza e derapata, assume un ruolo fondamentale nei moderni velivoli FBW, dotati di *flight control system full-authority*. Tali parametri, infatti, vengono utilizzati dai calcolatori di bordo per assolvere a funzioni essenziali, quali ad esempio la protezione dell'involuppo di volo e l'adeguamento dei guadagni dei sistemi di controllo al mutare delle condizioni di volo.

Inoltre la perdita delle informazioni relative ai dati aria, legata ad esempio al verificarsi di avarie dei sistemi di bordo, può comportare l'interruzione della missione ed il passaggio a modalità operative degradate, tali da consentire il rientro in sicurezza del velivolo alla base, con involucro di volo molto ristretto.

Da tali considerazioni risulta necessario dotare i sistemi ed i sensori dedicati alla determinazione dei dati aria di opportune ridondanze ed inoltre gli algoritmi di calcolo devono prevedere funzioni per individuare le avarie e riconfigurare il sistema.

Il lavoro svolto può esser sostanzialmente suddiviso in due parti.

Nella prima, dopo una breve descrizione dei requisiti e dei vantaggi legati all'impiego dei sistemi FBW, vengono descritti i modelli e gli algoritmi, sviluppati nell'ambito del progetto di ricerca, che consentono la stima dei parametri di volo, a partire dalle misure locali fornite dai sensori dati aria.

Quest'attività fa parte di un ampio progetto di ricerca, condotto a partire dal 1999 presso il Dipartimento di Ingegneria Aerospaziale dell'Università di Pisa, che ha come obiettivo lo studio di un sistema di controllo FBW dei comandi primari di volo, con attuazione idraulica, del tipo di quello implementato sull'addestratore militare di nuova generazione Aermacchi M346.

Nel presente lavoro è stata sviluppata una metodologia originale per l'identificazione delle avarie, mediante l'esame ed il confronto delle misure fornite dai sensori, la stima dei valori consolidati sia in condizione di piena operatività sia di avaria, ed infine la riconfigurazione del sistema secondo diverse modalità operative. Detta metodologia è stata implementata in ambiente *Stateflow*<sup>®</sup>, realizzando le modalità operative del sistema mediante lo schema di una macchina a stati.

In particolare il lavoro svolto si riferisce al caso concreto del sistema dati aria installato sull'addestratore M346: tale impianto utilizza quattro sonde

multi-funzione, installate sulla parte prodiera della fusoliera, capaci di allinearsi alla direzione locale del flusso.

Infine, è stato condotto uno studio approfondito dell'impatto delle possibili avarie sull'accuratezza di stima dei parametri di volo, prendendo in considerazione, mediante un approccio alla Monte Carlo, le caratteristiche statistiche dei segnali forniti dalle sonde.

Il contributo dato dalla presente tesi di dottorato, sia in termini di logiche di individuazione delle avarie, che di gestione delle ridondanze e riconfigurazione del sistema, può ritenersi del tutto originale. In letteratura, al momento, vengono descritte tecniche di gestione delle ridondanze, ad esempio relativamente alla determinazione della pressione a partire dai segnali forniti da quattro sonde, ma il problema oggetto della presente tesi è in realtà molto più complesso. Infatti, per quanto concerne la determinazione degli angoli di incidenza e derapata, è indispensabile disporre contemporaneamente dei segnali forniti da due sensori differenti. Le stime disponibili sono dunque sei: si tratta quindi di una gestione non-standard delle ridondanze ed in merito la letteratura fornisce scarsissime informazioni. Inoltre poiché è possibile il verificarsi di avarie sia dei sensori angolari che di pressione, occorre prevedere diverse modalità operative in funzione delle possibili combinazioni di guasto.

La seconda parte del lavoro è dedicata invece allo studio di tecniche alternative per la determinazione dei dati aria basate sull'impiego di sensori di tipo *Flush Air Data System*, FADS. Si tratta di sistemi di sensori non intrusivi, caratterizzati da un elevato numero di prese di pressione, disposti "a pelle" sulla parte prodiera della fusoliera.

Tale caratteristica rende i FADS particolarmente adatti ad applicazioni su velivoli da rientro spaziale, che devono esser privi di appendici soggette ad eccessivo riscaldamento in fase di rientro in atmosfera, velivoli non pilotati, che richiedono impianti a basso costo ed ingombri limitati, velivoli *stealth* poiché tali sistemi soddisfano l'esigenza di *low observability*.

La stima dei parametri di volo a partire da un numero elevato di segnali disponibili rappresenta un problema piuttosto complesso e comporta la necessità di monitorare tali segnali per individuare eventuali avarie.

Negli ultimi tempi si è sviluppato un notevole interesse, anche dal punto di vista industriale, nei confronti di tale tipologia di sensori: ne è testimonianza ad esempio il progetto europeo nEUROn, per lo sviluppo di un velivolo da combattimento non pilotato dalle accentuate caratteristiche *stealth*.

Nonostante questo interesse la letteratura risulta ancora povera di lavori sull'argomento. Il presente dottorato getta dunque le basi per una ricerca in tal senso. Dopo aver ipotizzato una possibile geometria del sensore-test, questa

è stata analizzata tramite studi di fluidodinamica computazionale per ottimizzarla ed ottenere un *database* campione, che lega le pressioni locali con le condizioni di volo.

Tale *database* è stato utilizzato per allenare le reti neurali, che costituiscono un approccio promettente per la ricostruzione dei parametri di volo.

## Ringraziamenti

Questi ringraziamenti sono rivolti ai miei professori Eugenio Denti e Roberto Galatolo per i preziosi consigli che mi hanno dato durante questa esperienza di Dottorato. Grazie al Prof. A. Salvetti per le "*chiacchierate*" sulla Dinamica degli Elicotteri.

Un ringraziamento particolare e più che doveroso va alla mia famiglia per gli infiniti incoraggiamenti che mi ha regalato in questi "*sudati*" anni di studio.

Grazie al "*Capo*" Francesco Schettini e a tutti i ragazzi del Laboratorio Fly-by-Wire che ho avuto modo di conoscere in questi cinque anni, o quasi, di lavoro!

Grazie a tutti gli amici del DIA.

Ed ora un grazie particolare a Massimiliano, agli amici di mensa, alle nuove compagne di "*scuola*" ed infine a Beatrice per gli spensierati momenti all'aria aperta.

# Indice

<b>1</b>	<b>I sistemi di controllo Fly-by-Wire</b>	<b>1</b>
1.1	Aspetti generali dei sistemi Fly-by-Wire Full Authority . . . . .	2
1.1.1	Definizioni . . . . .	2
1.1.2	Applicazioni e vantaggi del FBW . . . . .	3
1.1.3	Possibili rischi del FBW . . . . .	7
1.1.4	Affidabilità di un FCS/FBW . . . . .	8
1.2	Requisiti di un Flight Control System . . . . .	10
1.2.1	Compiti generali di un Flight Control System . . . . .	10
1.2.2	Architettura . . . . .	12
1.2.3	Funzioni operative del Flight Control System . . . . .	14
1.3	I sensori dati aria . . . . .	16
1.3.1	I sensori di pressione statica . . . . .	16
1.3.2	I sensori di angoli di flusso . . . . .	18
<b>2</b>	<b>Il Sistema Dati Aria</b>	<b>21</b>
2.1	Il sistema dati aria di riferimento . . . . .	23
2.2	Definizione del problema . . . . .	23
2.3	Caratterizzazione del campo aerodinamico . . . . .	26
2.4	Effetti di manovra sulle funzioni aerodinamiche . . . . .	26
2.5	Il database aerodinamico . . . . .	28
2.6	Avarie dei sensori dati aria . . . . .	29
2.7	Il software di simulazione . . . . .	29
2.8	La procedura di elaborazione . . . . .	32
2.8.1	Gestione delle ridondanze . . . . .	33
2.8.2	Algoritmi di monitoring angoli . . . . .	35
2.8.3	Algoritmi di monitoring pressioni . . . . .	36
2.8.4	Consolidamento delle avarie . . . . .	39
2.8.5	Il problema del Low Speed . . . . .	39



2.8.6	Algoritmi di voting . . . . .	41
2.8.7	Modi operativi della procedura di elaborazione dati aria . . . . .	42
2.9	L'ambiente Matlab/Stateflow . . . . .	45
2.9.1	Dal concetto di automa alla teoria delle macchine a stati finiti . . . . .	46
2.9.2	Stateflow: aspetti progetturali . . . . .	48
2.9.3	Applicazione del toolbox alle logiche di gestione delle ridondanze . . . . .	50
2.9.4	Vantaggi della programmazione in Stateflow . . . . .	55
2.10	Analisi dell'errore tramite simulazioni alla Monte Carlo . . . . .	57
2.11	Aspetti innovativi dell'argomento di ricerca trattato . . . . .	60
<b>3</b>	<b>Sviluppo di una sonda multi pressione basata su sistemi di tipo FADS</b>	<b>63</b>
3.1	Definizione della geometria . . . . .	64
3.2	Realizzazione della mesh tramite il software Gambit . . . . .	68
3.3	Analisi CFD e generazione del database aerodinamico . . . . .	75
3.3.1	Impostazione dei parametri di simulazione . . . . .	76
3.3.2	Visualizzazione ed analisi dei risultati . . . . .	78
3.4	Tecniche di intelligenza artificiale per la ricostruzione dei dati aria . . . . .	87
3.4.1	La metafora neurobiologica . . . . .	87
3.4.2	Campi d'applicazione e possibili problematiche . . . . .	89
3.4.3	Il principio di Occam . . . . .	90
3.4.4	Il neurone formale e le reti multi-livello . . . . .	91
3.4.5	Parametri di interesse e scelta delle prese di misura . . . . .	95
3.4.6	Valutazione degli errori commessi dalle reti a singola uscita . . . . .	101
3.4.7	Valutazione degli errori commessi dalle reti a doppia uscita . . . . .	103
3.5	Risultati generali . . . . .	103
3.6	Aspetti innovativi dell'argomento di ricerca trattato . . . . .	105
	<b>Conclusioni e sviluppi futuri</b>	<b>117</b>
	<b>Bibliografia</b>	<b>120</b>
	<b>Elenco delle Pubblicazioni</b>	<b>123</b>
	<b>Lista delle figure</b>	<b>126</b>
	<b>Elenco delle tabelle</b>	<b>130</b>
	<b>Elenco dei simboli</b>	<b>131</b>

# Capitolo 1

## I sistemi di controllo Fly-by-Wire

Negli ultimi decenni hanno trovato grande diffusione i sistemi automatici di controllo del volo con trasmissione elettrica dei segnali, denominati *Flight Control System* (FCS) *Fly-By-Wire* (FBW) *Full-Authority Full-Time*, che garantiscono un'affidabilità soddisfacente grazie ad un'architettura a molteplice ridondanza. Questi sistemi realizzano una serie di funzioni, quali controllo di configurazioni instabili, aumento della vita a fatica, prevenzione attiva del *flutter* e consentono ai moderni velivoli di raggiungere elevate prestazioni con un notevole grado di sicurezza.

Nel presente capitolo vengono definiti i requisiti che un sistema FCS/FBW deve soddisfare. In particolare si farà riferimento ad un moderno velivolo da addestramento militare trattando tali requisiti in modo essenzialmente qualitativo, evitando dunque di introdurre dati quantitativi che possono essere meglio definiti in una vera e propria specifica di progetto. I requisiti sono stati raggruppati in base ai componenti a cui essi si riferiscono (ad esempio *computer* di controllo del volo, attuatori).

Nel capitolo vengono trattati i principi su cui si fonda il progetto di un FCS, senza entrare nel dettaglio delle prestazioni richieste ai singoli componenti, ma riportando degli esempi di architettura del sistema o dei singoli componenti.

E' importante sottolineare che con l'introduzione dei sistemi automatici di controllo del volo di tipo FBW assumono un ruolo fondamentale le informazioni provenienti dai sensori di bordo per la determinazione dello stato del velivolo. Un sistema FBW non può prescindere dall'uso attivo ed integrato delle informazioni che provengono dai sistemi dei sensori, in particolare dai sensori dati aria.

## 1.1 Aspetti generali dei sistemi Fly-by-Wire Full Authority

### 1.1.1 Definizioni

Il sistema di controllo del volo (*Flight Control System*, FCS) è l'insieme di tutti i componenti che consentono il controllo di un velivolo, dallo *stick* di comando sino alle superfici mobili. Un *Automatic Flight Control System* (AFCS), secondo la definizione data in MIL-F-9490D<sup>[1]</sup>, è costituito dall'insieme dei componenti elettrici, meccanici ed idraulici che generano e trasmettono comandi automatici di controllo, in modo da assistere il pilota agendo sul controllo della traiettoria di volo, o sulla risposta della struttura del velivolo alle perturbazioni.

La funzione principale del FCS è quella di assicurare le opportune caratteristiche di stabilità e controllo all'interno dell'intero inviluppo di volo operativo, per ogni valore ammissibile di quota, velocità, fattore di carico e posizione del centro di gravità.

Un sistema FCS/FBW *Full Authority* è un sistema primario di controllo del volo con trasmissione elettrica dei segnali di comando, dove con primario si intende un sistema capace di controllare le superfici considerate essenziali per la stabilità del velivolo (piani di coda orizzontali, alettoni, timone, ...).

L'architettura di un moderno FCS/FBW è schematizzabile come in Fig. 1.1.

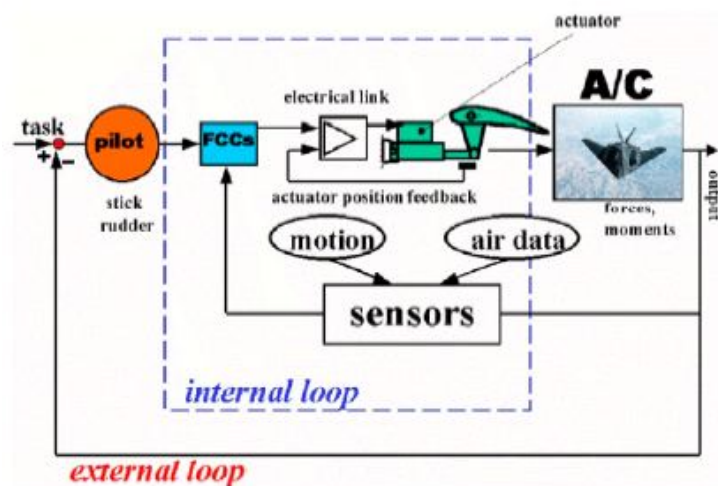


Figura 1.1: Schema di un sistema di controllo FBW.

Gli *input* del pilota, insieme agli *output* dei sensori del moto del velivolo, sono elaborati dai (*Flight Control Computer*, FCC), che inviano un opportuno comando agli attuatori delle superfici di controllo. Dunque un sistema FBW non può prescindere delle informazioni che provengono dal sistema dei sensori, elementi essenziali dunque del FCS.

Il pilota comanda il moto del velivolo piuttosto che la semplice deflessione delle superfici di controllo, ovvero controlla il velivolo mediante i FCC. Questi ultimi determinano l'entità degli spostamenti delle superfici di controllo per rispondere ai comandi, in modo veloce e ben smorzato all'interno dell'inviluppo di volo. Gli *input* provenienti dai comandi di volo sono tradotti in funzioni obiettivo: i FCC operano in modo da minimizzare l'errore tra queste funzioni e lo stato reale dell'aereo, misurato dai sensori inerziali e da quelli relativi ai dati aria.

Il pilota può essere considerato come l'elemento di chiusura del *loop* esterno di un sistema il cui *loop* interno è costituito dal FCS, a sua volta composto dai seguenti sottosistemi:

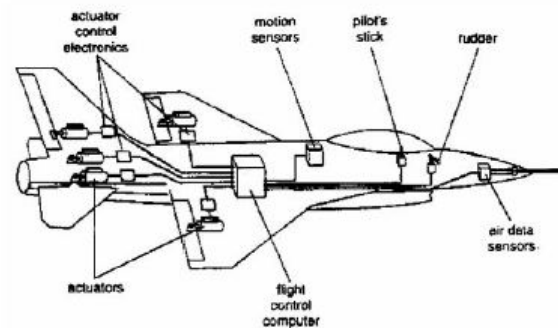
- computer(FCC) che elaborano il comando del pilota attraverso il controllo continuo della dinamica del velivolo e l'applicazione di opportune leggi di controllo;
- sensori che acquistano i dati aria e misurano le componenti di accelerazione lungo i tre assi dell'aereo e quelle di velocità angolare attorno ad essi;
- attuatori delle superfici di controllo e dei relativi sensori di posizione;
- cavi elettrici che consentono la trasmissione dei segnali e dei comandi tra i vari componenti.

L'insieme dei vari elementi è schematizzato in Fig. 1.2, tratta da<sup>[2]</sup>.

### 1.1.2 Applicazioni e vantaggi del FBW

Il FCS/FBW realizza contemporaneamente due aspetti, di per sé in conflitto: stabilità e ottimo controllo del velivolo.

Il FCS/FBW risulta dunque di fondamentale importanza su aerei da combattimento di ultima generazione: in questi casi le prestazioni richieste portano a configurazioni naturalmente instabili che, se non adeguatamente supportate da un sistema di controllo automatico, raddoppierebbero l'entità di un disturbo in tempi molto brevi (tipicamente dell'ordine del decimo di secondo). Le forze aerodinamiche agenti sulle superfici di controllo sono così grandi e così velocemente variabili che diventa impossibile per il pilota mantenere lo



**Figura 1.2:** Componenti fondamentali di un FCS/FBW .

sforzo richiesto sui comandi per un periodo prolungato.

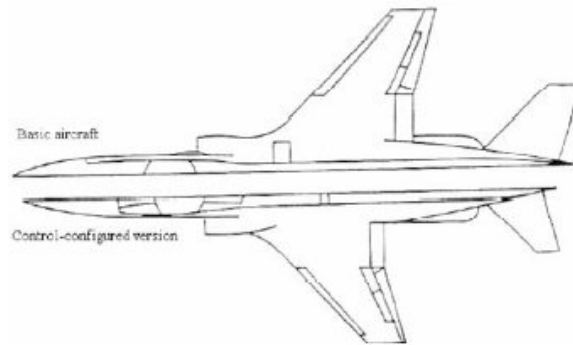
Inoltre in un velivolo stabile la coda, in molte condizioni di volo, dà un contributo negativo (coda deportante) che l'ala compensa fornendo più portanza di quanto non sia necessario per sostenere il velivolo. Al contrario in un aereo instabile la coda fornisce generalmente un contributo positivo alla portanza complessiva. Nelle configurazioni instabili si possono perciò impiegare superfici di stabilizzazione più piccole, ridurre le dimensioni dell'ala e di conseguenza avere maggiore efficienza aerodinamica (ovvero minore resistenza) e minor peso in alternativa o in aggiunta all'incremento di prestazioni (minori distanze di decollo e di atterraggio, minori raggi di virata, miglior risposta alla raffica). Un aereo il cui volo è controllato dal pilota mediante l'ausilio continuo del computer è detto *Control Configured Vehicle* (CCV).

La Fig. 1.3 e la Fig. 1.4 sono tratte da<sup>[2]</sup> e<sup>[3]</sup> ed illustrano quanto descritto in precedenza.

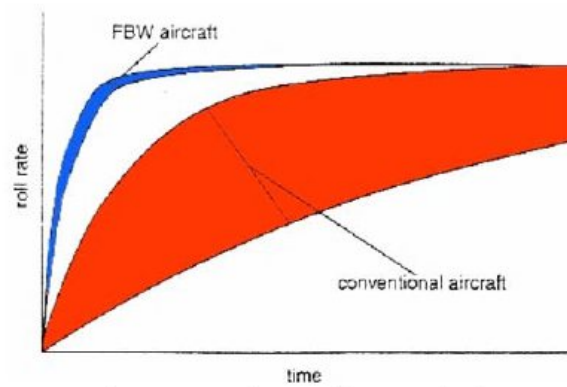
Oltre alla gestione ottimale delle superfici primarie di controllo, ai FCC si può anche affidare la cosiddetta funzione di protezione dell'involuppo (*care-free manoeuvring*). Si tratta di una continua analisi di prossimità ai limiti dell'involuppo di volo e della modifica automatica del comando del pilota, al fine di impedire che il velivolo assuma un assetto inaccettabile, avvicinandosi allo stallo o effettuando manovre che determinano il superamento dei limiti strutturali.

Ai FCC possono esser affidate funzioni di controllo automatico di:

- angolo di deflessione dei *leading edge* e *trailing edge flap* in manovra, al decollo ed in atterraggio, in modo da ottenere la minima resistenza per ogni valore della portanza e del numero di Mach;



**Figura 1.3:** Effetto scala di un CCV.



**Figura 1.4:** Confronto di prestazioni.

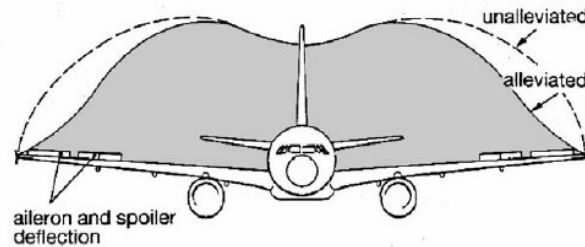
- angolo di freccia alare per aerei a geometria variabile;
- direzione della spinta del motore (*thrust vectoring*) per velivoli dotati di ugelli orientabili;
- angolo di deflessione di particolari superfici aerodinamiche localizzate, volte ad aumentare le prestazioni (*direct force control*)<sup>[4]</sup>.

Inoltre è possibile attenuare la risposta elastica della struttura ai carichi da raffica, mediante le superfici di controllo standard dell'aereo (elevatore, timone di coda, alettoni), la cui azione viene definita e coordinata dal FCS/FBW, in risposta a segnali inviati da accelerometri localizzati. La durata a fatica può essere addirittura raddoppiata per aerei di notevoli dimen-

sioni<sup>[4]</sup>.

Un esempio di installazione su aerei commerciali di un simile sistema si ha nell'Airbus A320, vedi Fig. 1.5 tratta da<sup>[5]</sup>.

L'applicazione di questa tecnica ad aerei da combattimento di ridotte di-



**Figura 1.5:** *Gust alleviation control.*

mensioni sembra però improbabile, sia perché per questi velivoli sono più importanti le sollecitazioni dovute ai carichi di manovra rispetto a quelle dovute ai carichi di raffica, sia perché le frequenze dei modi strutturali sono più elevate e quindi richiedono migliori capacità di controllo. Una tecnica analoga, diretta alla stabilizzazione dei modi strutturali e alla prevenzione attiva del *flutter* (vibrazioni aeroelastiche autosostenute che possono portare ad un cedimento strutturale) permette un ulteriore rilassamento della rigidità strutturale e notevoli guadagni in termini di peso del velivolo, a parità di margine ammesso sulla velocità di *flutter*. Anche in questo caso però è improbabile l'applicazione ad aerei da combattimento a causa delle elevate frequenze proprie dei modi strutturali e dello scarso spazio (piccoli spessori delle superfici portanti) disponibile per l'installazione dei sistemi di controllo. E' importante sottolineare che l'adozione di FCS/FBW consente l'incremento delle proprietà *stealth* di un velivolo (ovvero della capacità di sfuggire all'intercettazione radar) per le ragioni elencate di seguito:

- riduzione dell'estensione delle superfici di controllo tradizionali, che notoriamente costituiscono uno dei maggiori componenti radar-riflettenti del velivolo;
- possibilità di adottare configurazioni più favorevoli, ma intrinsecamente meno stabili (ad esempio il Northrop B2 in Fig. 1.6 ed il Lockheed F117 in Fig. 1.7, che hanno differenti filosofie di *stealthness*);
- razionalizzazione dell'uso delle superfici di controllo e conseguente riduzione dell'effetto riflettente delle punte.



**Figura 1.6:** *Northrop-Grumman B2 Spirit.*



**Figura 1.7:** *Lockheed F117 Nighthawk.*

### 1.1.3 Possibili rischi del FBW

Il controllo ideale di un velivolo instabile richiederebbe tempi di reazione virtualmente nulli, tra l'istante in cui una correzione si rende necessaria e l'istante in cui essa, di fatto, viene attuata mediante un movimento della superficie di controllo. In realtà, i ritardi sono sempre presenti in qualunque forma di controllo: in particolare nei FBW analogici o digitali sorgono dalla necessità di condizionare il segnale, filtrare il rumore ed elaborarlo. Dal momento che i sistemi digitali operano discontinuamente, occorre verificare che i ritardi non degradino la stabilità dell'intero sistema. Ciò vale in particolare per la delicata fase di disconnessione di un canale riconosciuto in avaria, ed il relativo transitorio nelle attività di *voting*, ovvero di gestione delle informazioni ridondanti : tale situazione è tanto più critica quanto più è naturalmente instabile la configurazione cui si applica il controllo.

Sono importanti la sensibilità e l'esperienza del pilota di un velivolo FBW, ma anche una corretta progettazione dell'intero FCS, al fine di ridurre ed al limite eliminare fenomeni di *Pilot Induced Oscillations* (PIO), che si presentano



in CCV, in fase di approccio ed atterraggio. La principale causa riconosciuta di PIO<sup>[6]</sup> non è un difetto del *software* o dell'architettura digitale di per sé, ma la disarmonia tra l'azione del pilota e la reazione che questi si attende dal velivolo. Al pilota pervengono informazioni confuse od in conflitto ed il suo guadagno può diventare così elevato da rendere il *loop* esterno del sistema Fig. 1.1 dinamicamente instabile. Tale disarmonia può essere causata da una o più concause eliminabili, almeno in parte, con un progetto accurato del FCS e delle leggi di controllo: eccessiva sensibilità della barra di comando, eccessivi ritardi di fase del sistema, elevate non-linearità, risposte poco smorzate o instabili.

Nei limiti delle prestazioni che si vogliono conseguire e delle configurazioni che si vogliono esplorare, la differenza tra il comportamento del velivolo sotto le leggi di controllo, e quello sotto le leggi degradate (corrispondente al verificarsi di una o due *failure*, e più vicino al comportamento naturale) deve essere dunque limitata in modo da evitare il disorientamento al pilota e l'eccessivo carico di lavoro. Inoltre, quanto più il comportamento controllato differisce da quello naturale, tanto più i servo-controlli risultano sollecitati (massime escursioni e velocità di attuazione, fatica)<sup>[5]</sup>.

Per velivoli commerciali dovrebbe essere considerato anche il *comfort* dei passeggeri. Il passaggio ai modi controllati, sia in automatico che effettuato dal pilota, dovrebbe essere graduale, ed il comportamento del velivolo in accordo con le sensazioni e le risposte che il pilota si aspetta, con leggi di controllo pienamente efficienti o con modi degradati.

#### 1.1.4 Affidabilità di un FCS/FBW

In generale un FCS è costituito da componenti ridondanti il cui numero dipende dal tipo di velivolo e dagli obiettivi di affidabilità richiesti. Inoltre, il numero di motori influenza l'architettura del FCS in quanto determina il numero di impianti idraulici ed elettrici.

Nei velivoli moderni si vanno affermando sistemi in grado di sopportare due cedimenti gravi (o avarie) e conservare sufficiente capacità operativa, al fine di soddisfare i requisiti di sicurezza e garantire ancora determinati margini di manovra. Per cedimento grave s'intende *cessazione della capacità di un componente di svolgere le funzioni richieste entro i limiti stabiliti dalla specifica*. In altre parole, il FCS/FBW deve garantire gli stessi livelli di sicurezza ed integrità del sistema meccanico che sostituisce. La sicurezza è specificata in termini di probabilità che si verifichi un'avaria nel FCS tale da comportare la perdita del velivolo. Le normative aeronautiche impongono che la probabilità che si verifichi un guasto catastrofico (ovvero un guasto che può provocare un danno ad un numero notevole di persone) debba essere inferiore a  $10^{-9}$  e  $10^{-7}$

per ora di volo, rispettivamente per un velivolo civile ed uno militare,<sup>[2][7]</sup> . Poiché la componentistica attuale non è in grado di assicurare con singoli elementi un simile livello di prestazioni, è previsto un certo grado di ridondanza nel sistema: si utilizzano più canali dotati di analogo equipaggiamento *hardware/software*, così da svolgere simultaneamente le medesime funzioni. Inoltre è previsto un monitoraggio continuo del sistema, al fine di rilevare eventuali discrepanze tra i diversi canali ed escludere tramite un apposito algoritmo quei dati che sono in disaccordo con gli altri.

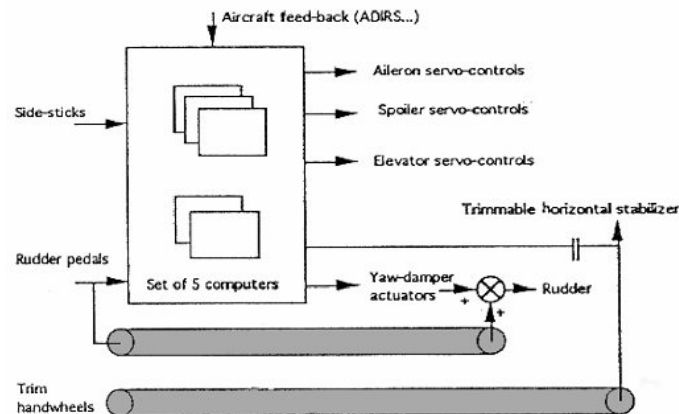
Si distinguono in particolare due filosofie diverse per il FBW: una devolve completamente il controllo a componenti elettrici/elettronici (FBW *Full Authority*), per cui non possono essere assolutamente accettate avarie che rendano inutilizzabile il FCS; l'altra prevede la presenza di un *backup* meccanico e/o analogico di tipo tradizionale, almeno limitatamente ai comandi fondamentali, al quale viene affidato il controllo minimo del velivolo in caso di avaria totale del sistema primario (ad esempio *Boeing 777*, *Airbus A320*, *A330*, *A340*). Il secondo approccio è adottato prettamente in campo civile, perché in questo tipo di aerei non vengono richieste prestazioni così elevate da rendere di fatto impossibile il controllo del velivolo senza l'aiuto del FCS/FBW, come invece avviene nei velivoli da combattimento. La necessità di conservare una configurazione controllabile anche solo per mezzo di semplici connessioni meccaniche di *backup* limita, di fatto, le prestazioni teoricamente raggiungibili mediante l'uso di un sistema principale di controllo di tipo FBW.

In Fig. 1.8 è riportato lo schema della soluzione FBW adottata da *Airbus* per l'A320: essa prevede cinque *computers* simultaneamente attivi, ciascuno dei quali, per sicurezza, è diviso in due canali fisicamente separati: il primo, il canale di controllo, è continuamente monitorato dal secondo, il canale di monitoraggio<sup>[5]</sup> .

Il FCS primario è affiancato da un *backup* di tipo meccanico sullo stabilizzatore orizzontale e sul timone di coda (*rudder*).

La manovrabilità in rollio è in questo caso assicurata, entro certi limiti, dagli effetti aerodinamici secondari che si generano in seguito alla deflessione del *rudder* (rollio indotto).

Per quanto riguarda i livelli di affidabilità dei singoli componenti, attualmente, per un sistema FBW *Full Authority* potrebbe esser richiesta anche una quadruplice ridondanza degli elementi essenziali del FCS<sup>[4]</sup> , il che porta a soluzioni del tipo di quella schematicamente mostrata in Fig. 1.9. Le quattro linee di controllo sono totalmente indipendenti e la probabilità che due vengano meno contemporaneamente è estremamente remota. Una singola avaria su una delle linee viene scavalcata dal buon funzionamento delle altre: la li-



**Figura 1.8:** FCS/FBW con *backup* meccanico per l'Airbus A320.

nea difettosa viene identificata e sconnessa mediante un confronto incrociato dei dati e dal *voting*, che restituisce un unico valore consolidato.

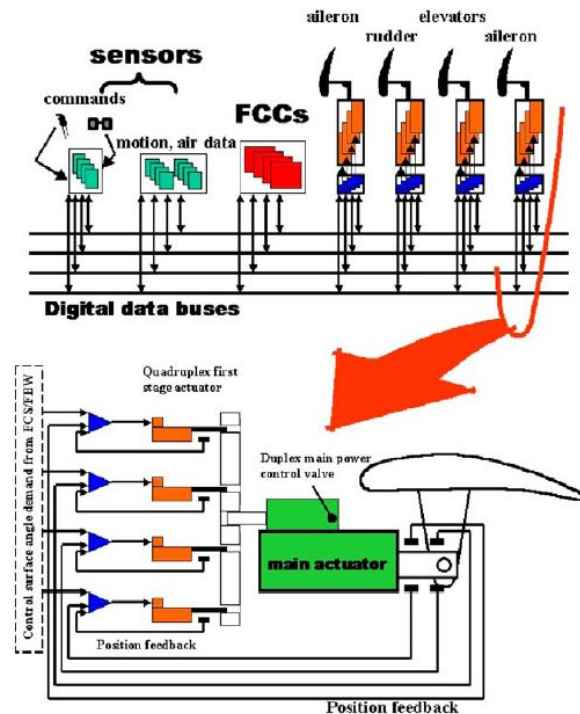
Allo scopo di evitare la cosiddetta *common-mode failure* (causa comune di guasto), per cui nello stesso istante tutti e quattro i canali restituiscono un dato errato, occorre prevedere il contemporaneo verificarsi di determinate cause ambientali, oltre ad adottare opportune procedure di progettazione. È necessario, infatti, cautelarsi da errori comuni di programmazione, quali quelli, ad esempio, che possono verificarsi qualora il *software* sia lo stesso per tutti i canali. È comunque molto difficile, se non impossibile, dimostrare con certezza l'assenza di errori nei codici. Per questo motivo, le autorità civili richiedono l'adozione nel FCS di sistemi diversi (anche se analoghi) in ridondanza: diversi sensori, differenti microprocessori, e *software* realizzati da gruppi diversi di programmatori<sup>[2]</sup>.

## 1.2 Requisiti di un Flight Control System

### 1.2.1 Compiti generali di un Flight Control System

Il FCS deve assicurare le caratteristiche di controllo e stabilità richieste all'interno dell'intero inviluppo di volo operativo, per ogni valore di quota, velocità, fattore di carico e posizione del centro di gravità.

Il FCS comanda le superfici di controllo (Fig. 1.10) ed assicura lo smorzamento sui tre assi del velivolo; per ordine di importanza il controllo si distingue in:

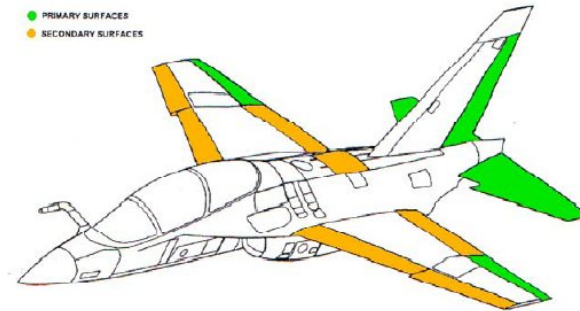


**Figura 1.9:** Ridondanza di un FCS/FBW e delle sue parti.

- **Primario:** lungo gli assi di beccheggio, rollio ed imbardata, generalmente assicurato dalle superfici di controllo dei piani orizzontali di coda, dagli alettoni e dal timone;
- **Secondario:** realizzato tramite ipersostentatori di bordo d'attacco e di bordo d'uscita e tramite gli aerofreni, che assicurano rapide decelerazioni ed un accurato controllo della velocità dell'aereo. Gli ipersostentatori garantiscono il raggiungimento di elevati valori di portanza: la deflessione di quelli installati sul bordo d'attacco è regolata in modo da ottimizzare le caratteristiche aerodinamiche dell'aereo in base al valore dell'angolo di incidenza, del numero di Mach e dell'angolo di deflessione degli ipersostentatori di bordo di uscita.

Il FCS deve essere progettato in modo da assicurare le prestazioni richieste minimizzando il carico di lavoro sul pilota.

Il FCS deve conseguire gli obiettivi generali sull'affidabilità assegnati per l'intero velivolo, con una probabilità di perdita dell'aereo dovuta ad un cedimento grave (*failure*) del FCS non superiore a 1 su  $10^7$  /  $10^9$  ore di volo



**Figura 1.10:** Superfici di controllo di un velivolo intercettore leggero.

(valori tipici assunti rispettivamente per velivoli militari e civili), escludendo i guasti ai generatori elettrici ed idraulici.

Il FCS deve risultare compatibile con tutte le interfacce associate e deve assicurare, sia a terra che in volo, le prestazioni richieste in specificate condizioni ambientali, normali ed estreme. Devono essere inoltre soddisfatti i requisiti generali di integrità, bassa interdipendenza tra i componenti, basso costo e facilità di manutenzione.

### 1.2.2 Architettura

I FCS adottati sui principali velivoli da combattimento di ultima generazione (Rafale, Eurofighter, F22 Raptor) sono del tipo *Full-Authority*, privi di sistemi di *back-up* analogico/meccanico e sono da considerarsi come lo stato dell'arte a livello di controllo automatico del velivolo.

Il FCS comprende tutti i componenti necessari a realizzare le funzioni previste entro i limiti operativi, assegnati in termini di prestazioni, affidabilità ed integrità.

I vari sottosistemi possono essere raggruppati secondo quattro principali aree funzionali: *cockpit* (limitatamente alle funzioni correlate al FCS), sensori, *computers*, sistema di attuazione.

**Cockpit:** gli elementi del *cockpit* correlati alle funzioni del FCS sono la barra di comando, i pedali, i dispositivi per la sensibilità artificiale, gli attuatori di *trim*, i pannelli di controllo.

**Sensori:** il sottosistema dei sensori consente di acquisire le informazioni sullo stato del velivolo inteso come sistema dinamico e comprende:

- i sensori di flusso e quelli realizzati mediante sonde esterne al velivo-

lo (sottosistema dei dati aria e della valutazione della direzione della corrente);

- i sensori inerziali che forniscono le velocità angolari e le accelerazioni lineari secondo i tre assi del velivolo;
- i sensori disposti all'interno del velivolo (sottosistema dei dati inerziali relativi al moto del velivolo);
- i sensori dei sottosistemi secondari che forniscono eventuali informazioni accessorie sullo stato dell'aereo (ad esempio, ripartizione del peso sui carrelli).

**Computer:** i *computer* per il controllo del volo sono il cuore del FCS, in quanto acquisiscono ed elaborano i parametri inviati dai sensori e dagli organi di comando del pilota secondo le leggi di controllo realizzate, monitorano il sistema, individuano le eventuali avarie transitorie o permanenti, forniscono ai vari sottosistemi dati corretti e consolidati ed infine inviano i necessari segnali agli attuatori delle superfici di controllo.

**Sistema di attuazione:** il sistema di attuazione consente di tradurre i comandi forniti dai *computer* in movimenti delle superfici di controllo. Esso si può dividere ulteriormente in:

- sottosistema di attuazione delle superfici di controllo primarie (piani di coda orizzontale, alettoni, timone), essenziali per garantire la stabilità ed il controllo del velivolo;
- sottosistema di attuazione delle superfici di controllo secondarie (ipersostentatori di bordo d'attacco e di bordo di uscita, aerofreni), la cui perdita parziale o totale penalizza ma, in generale, non compromette la realizzazione della missione;
- sottosistema dei sensori di posizione degli attuatori, necessari per inviare ai FCC le informazioni relative alla deflessione delle superfici. Si deve assicurare l'irreversibilità del comando di attuazione e garantire i necessari smorzamenti, al fine di prevenire fenomeni di *flutter* o di instabilità dinamica.

In generale, per garantire elevati livelli di affidabilità e sicurezza, si fa in modo che il FCS mantenga una completa capacità operativa anche in caso di duplice avaria (*two fail operative/fail safe capability*); questo requisito può comportare nei velivoli moderni una architettura caratterizzata da quadruplice ridondanza.

### 1.2.3 Funzioni operative del Flight Control System

Il FCS deve essere capace di svolgere in modo efficace alcune funzioni, considerate principali ed altre addizionali. Quelle principali sono raggruppate in funzioni manuali, automatiche e di navigazione avionica. Quelle addizionali possono essere ad esempio impiegate per le operazioni di puntamento e tiro.

Nel seguito si elencano le funzioni principali svolte dal FCS.

Funzioni per il sistema di controllo manuale del volo (*Manual Flight Control System Functions*):

- Aumento della stabilità e del controllo: questa funzione consiste nel garantire un'adeguata stabilità ed un efficace controllo (manovrabilità) su tutti e tre gli assi;
- Protezione dell'involuppo: consente di prevenire il superamento dei limiti ammessi dell'involuppo di volo;
- Trimmaggio: permette di assicurare l'equilibrio delle forze e dei momenti aerodinamici agenti sul velivolo in una certa condizione di volo senza che il pilota agisca sui comandi;
- Controllo della configurazione: questa funzione ha l'obiettivo di regolare, oltre alla deflessione delle superfici primarie, la deflessione dei *flap* di bordo di attacco, di bordo di uscita e l'estensione dell'aerofreno. I *flap* di bordo di uscita e l'aerofreno sono comandati dal pilota, mentre i *flap* di bordo di attacco sono opportunamente gestiti dai FCC;
- Controllo della sterzata con il carrello anteriore: deve consentire un controllo preciso e stabile della sterzata al suolo per tutte le velocità di rullaggio;
- Controllo del sistema: deve assicurare la gestione dei segnali ridondanti, delle avarie e dei vari modi operativi del FCS.

Funzioni per il sistema di controllo automatico del volo (*Automatic Flight Control System functions*):

- Autopilota
- Direttore di volo: questa funzione fornisce al pilota delle indicazioni relative all'assetto che egli deve mantenere al fine di svolgere un certo *task* di pilotaggio (ad es. mantenimento della prua o del sentiero di avvicinamento).

Funzioni per il sistema avionico di navigazione:

- Calcolo dei dati per la navigazione aerea: lo scopo di questa funzione è di fornire all'avionica le informazioni necessarie alla navigazione;
- Funzione di interfaccia: consente l'interazione tra il FCS e gli altri sistemi del velivolo, come quello che controlla il funzionamento del motore.

Queste funzionalità devono essere pienamente garantite quando il sistema è al massimo delle proprie capacità operative ed assicurate secondo modalità e limiti prefissati in caso di avarie, effettive oppure simulate dal pilota a scopi addestrativi.

Le funzioni sopra elencate sono realizzate, quindi, secondo diversi *modi operativi* che corrispondono ai gradi di funzionalità del FCS.

Tipicamente si individuano quattro modi operativi:

- Modo Normale: il FCS è pienamente operativo.
- *Alpha-fail mode* (modo di avaria del segnale di incidenza, capacità operativa ristretta): questa modalità prevede di non disporre delle informazioni relative all'angolo di incidenza aerodinamica.
- *Fixed-gain mode* (modo con guadagni fissi del sistema di controllo): questa modalità prevede di assolvere le funzioni di controllo con le limitazioni derivanti dalla mancanza dei segnali relativi alle pressioni totali e statiche.
- *Digital Back-Up* (modo di riserva): questo modo operativo prevede l'impiego di un *software* di controllo di riserva (*back-up*) completamente indipendente da quello principale.

Il FCS deve consentire il controllo completo del velivolo, in condizioni di piena operatività, riducendo il carico di lavoro sul pilota.

Il controllo longitudinale viene tradizionalmente ottenuto mediante la deflessione delle superfici di controllo del piano di coda orizzontale.

Il FCC interviene su tali superfici al fine di ottenere variazioni di assetto in beccheggio e garantire l'equilibrio del velivolo nel piano longitudinale. Gli impennaggi orizzontali consentono anche di smorzare i moti propri del velivolo e compensare la variazione di momento aerodinamico dovuta all'estrazione degli aerofreni o alla deflessione degli ipersostentatori.

Il controllo laterale è in genere ottenuto mediante la deflessione degli alettoni mentre quello direzionale è effettuato mediante la deflessione del timone.

Il *rudder* ha anche la funzione di aumentare lo smorzamento dinamico dei



moti nel piano latero-direzionale.

Il controllo in portanza è ottenuto per mezzo della deflessione degli ipersostentatori di bordo di attacco e di bordo di uscita. In particolare, la configurazione di quelli al bordo di uscita è selezionabile da parte del pilota fra un certo numero di alternative possibili, a seconda della fase di volo, mentre quella degli ipersostentatori di bordo di attacco è automatica e regolata dai FCC. L'estrazione degli aerofreni, utili per ridurre la velocità dell'aereo in tempi molto brevi, può essere comandata dal pilota o dal sistema di controllo.

### 1.3 I sensori dati aria

Con l'introduzione dei sistemi automatici di controllo di volo di tipo FBW, assumono importanza fondamentale le informazioni provenienti dai sensori di bordo per la determinazione dello stato del velivolo.

Un sistema FBW non può prescindere dall'uso attivo ed integrato delle informazioni che provengono dal sistema dei sensori, che sono perciò elementi essenziali del FCS.

In particolare i sensori dati aria consentono la misurazione della pressione statica ambiente, corrispondente alla quota di volo, e della pressione totale. Da tali pressioni viene calcolato il numero di Mach, parametro indispensabile per la variazione dei guadagni delle leggi di controllo programmate all'interno dei *computer* di bordo. Inoltre il sistema dati aria deve provvedere alla misurazione dell'assetto del velivolo rispetto alla direzione del flusso aerodinamico della corrente indisturbata, cioè degli angoli di incidenza e di derapata: questi angoli sono fondamentali per l'implementazione di funzioni quali la protezione dallo stallo e la deflessione automatica di superfici di controllo secondarie per l'ottimizzazione dell'aerodinamica del velivolo.

Di seguito sono riportate le principali tipologie di sensori di pressione statica, angolo di flusso, accelerometrici e giroscopici usati in ambito aeronautico, spiegandone brevemente i principi di funzionamento ed evidenziandone le problematiche per il loro uso.

#### 1.3.1 I sensori di pressione statica

Nella pratica risulta impossibile posizionare i sensori dati aria in modo da effettuare delle misurazioni nel campo aerodinamico indisturbato. Gli inevitabili effetti di perturbazione del flusso dovuti ai vari assetti della fusoliera e alle manovre influenzano le misure dei i sensori.

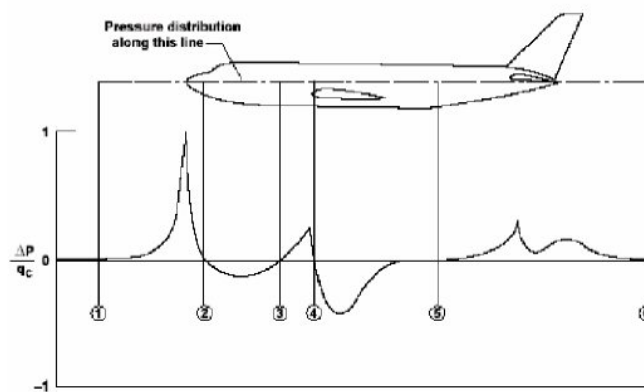
Le misure di pressione statica possono essere rilevate tramite tubi di *Pitot* statici o mediante più sonde di pressione montate sul velivolo.

Il campo di pressione statica attorno alla fusoliera è funzione della velocità, della quota, degli angoli di incidenza e derapata, ma anche della configurazione del velivolo.

La rotazione od estrazione di superfici mobili (*flap*, *aileron*, *nose droop*, *rudder*, ...), la presenza o meno dei carrelli di atterraggio, nonché le prese d'aria del motore, influenzano in maniera determinante il flusso aerodinamico in prossimità delle sonde. Dunque questi sensori lavorano su un flusso disturbato dalla presenza del velivolo. L'accuratezza nella misurazione della pressione statica ambiente dipende quindi, oltre che da caratteristiche geometriche e costruttive delle sonde stesse, anche dal posizionamento di esse sul velivolo. Anche se alcuni di questi aspetti possono essere studiati con prove in galleria aerodinamica, non si può prescindere da una fase di calibrazione in volo.

In Fig. 1.11 è mostrata la distribuzione di pressione lungo l'asse longitudinale di un tipico velivolo da trasporto<sup>[1]</sup>.

Il valore  $\Delta P$ , adimensionalizzato con la pressione dinamica  $q_c$ , rappresenta



**Figura 1.11:** Distribuzione di pressione attorno ad un velivolo da trasporto.

la differenza tra la pressione statica locale misurata e la pressione statica del flusso indisturbato a monte del velivolo. I punti numerati da 1 a 6 individuano i possibili posizionamenti delle sonde sul velivolo.

In volo subsonico generalmente si preferiscono le posizioni 2 e 3 in quanto la posizione 4 risente in maniera decisiva dell'estrazione dei *flaps*, mentre la 5, usata in qualche soluzione alternativa, è tuttavia troppo distante dal *cockpit* e quindi comporta dei problemi legati all'avionica.

Per velivoli supersonici, la soluzione *nose boom* (posizione davanti alla fusoliera e più vicina possibile a 1) non risentendo delle perdite dovute all'onda d'urto, è senz'altro la più usata.

In generale la migliore posizione delle sonde è determinata confrontando le diverse misurazioni rilevate con una sorgente di riferimento.

Alcuni *nose boom* e sonde di pressione statiche hanno forme disegnate appositamente per compensare gli errori di posizionamento.

Un problema aggiuntivo è costituito dalla considerazione che la distribuzione di pressione intorno al velivolo varia in funzione delle condizioni di volo. E' dunque necessaria una calibrazione in diversi punti dell'involuppo.

### 1.3.2 I sensori di angoli di flusso

Solitamente vengono utilizzati tre tipi di sensori di flusso locale<sup>[8]</sup> : *pivoted vane*, *differential pressure tube* e *null-seeking pressure tube*.

La sonda tipo *pivoted vane* si allinea automaticamente con il flusso a seguito del momento dovuto alle alette montate sulla sonda stessa (Fig. 1.12); dopo l'allineamento, un trasduttore di misura angolare rileva la direzione del flusso. Se montate su un supporto tipo asta vanno incontro ad errori dovuti alla distorsione del flusso causato dall'asta e alla sua flessione in seguito ai carichi aerodinamici. Alle basse velocità risultano essere più sensibili rispetto agli altri tipi di sensori, ma sono soggette a una più facile rottura.

I sensori di tipo *differential pressure tube* (Fig. 1.13) sono generalmente a

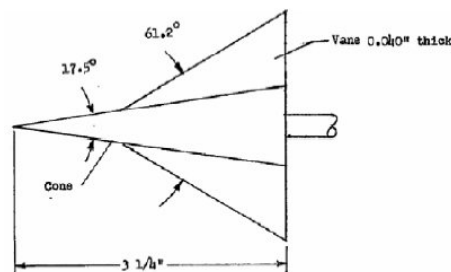


Figura 1.12: Sensore *pivoted vane*.

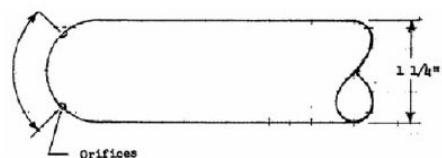


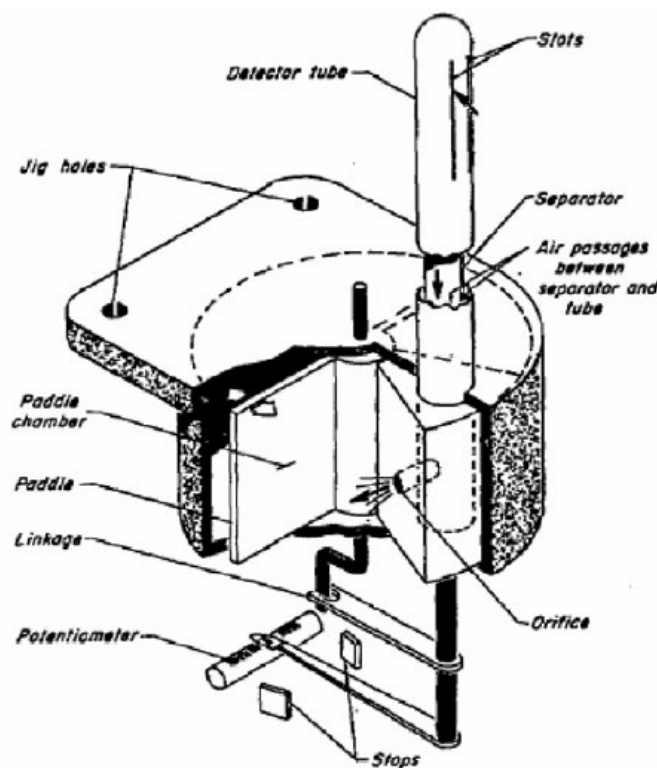
Figura 1.13: Sensore a testa sferica.

testa sferica o piramidale, con due prese di pressione per la misurazione dell'angolo rispetto a ogni asse.

Le differenze di pressione sono convertite in misure d'angolo con opportuna calibrazione.

Simili ai precedenti, i sensori *null-seeking pressure tube* (Fig. 1.14) hanno un servomeccanismo per allineare il sensore al flusso e, tramite un trasduttore d'angolo, misurano la direzione del flusso locale.

Come per i sensori di pressione statica, il posizionamento dei sensori di flusso

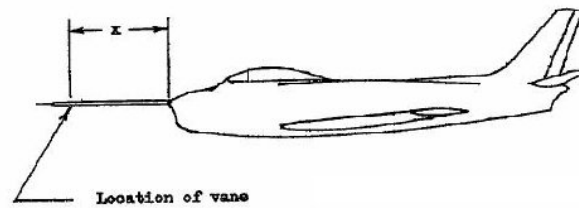


**Figura 1.14:** Sonda *null-seeking pressure tube*.

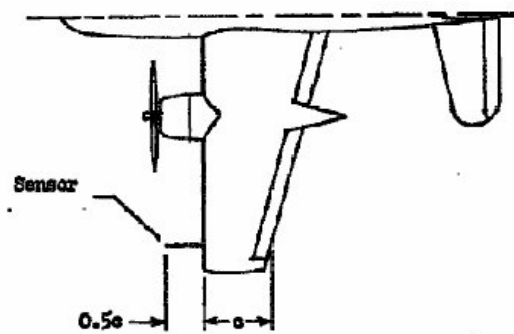
sul velivolo influenza in maniera determinante l'accuratezza delle rilevazioni. I sensori misurano, in effetti, l'angolo di flusso locale piuttosto che l'angolo di incidenza e derapata del velivolo. Servono quindi delle fasi di calibrazione sia sulle sonde isolate (generalmente in galleria del vento), sia relativamente all'errore dovuto al posizionamento sul velivolo. Questo ultimo errore è influenzato da numerosi aspetti quali, numero di Mach, condizioni di carico aerodinamico e configurazione del velivolo (estrazione delle superfici mobili,

presenza o meno dei carrelli di atterraggio, ecc).

Tipiche installazioni dei sensori di flusso sul velivolo sono rappresentate nelle figure seguenti tratte da<sup>[9]</sup>.



**Figura 1.15:** Installazione tipo *nose boom*.



**Figura 1.16:** Installazione tipo *wingtip*.



**Figura 1.17:** Installazione in fusoliera.

# Capitolo 2

## Il Sistema Dati Aria

Un sistema dati aria è l'insieme degli apparati del velivolo dedicati alla stima della pressione statica  $P_{sa}$ , della pressione totale  $P_t$ , del numero di Mach di volo  $M_\infty$ , degli angoli di incidenza  $\alpha$  e derapata  $\beta$  a partire da misure locali. La rilevazione fisica di tali misure è effettuata da un sistema di sensori, installati nella parte prodiera della fusoliera o sulle ali.

La derivazione dei parametri di volo a partire dalle misure dei sensori, viene di solito affidata ad un algoritmo, denominato procedura di elaborazione, implementato nei FCC del FCS o in unità di calcolo dedicate.

Durante una generica missione, il verificarsi di una avaria di uno o più sensori, può causare la perdita, parziale o totale, delle misure da esso fornite. Per questo motivo i requisiti di affidabilità richiesti ai moderni FCS/FBW, riportati in<sup>[2]</sup> e<sup>[7]</sup>, prevedono l'implementazione di opportuni gradi di ridondanza dei singoli componenti e quindi lo sviluppo di logiche di adeguata robustezza, per la gestione delle ridondanze stesse, l'identificazione delle avarie (*failure*) e la riconfigurazione del sistema.

Tutto ciò comporta la presenza di diverse modalità operative.

Inoltre, la scelta della tipologia e dei punti di installazione delle sonde costituisce un problema delicato, in quanto è necessario garantire che il sistema dati aria sia operativo per tutti gli assetti ( $\alpha$ ,  $\beta$ ) ammissibili nell'involuppo di volo considerato. Per velivoli ad elevate prestazioni il campo di variazione degli assetti è estremamente ampio e può esser difficile evitare condizioni in cui la sonda si venga a trovare nella scia di altre parti del velivolo stesso.

Il capitolo illustra i modelli ed i metodi sviluppati durante il presente dottorato per l'integrazione dei dati aria nell'FCS di un moderno velivolo FBW. Quest'attività fa parte di un ampio progetto di ricerca, condotto dal Dipartimento di Ingegneria Aerospaziale (DIA) di Pisa, che ha come obietti-

vo lo studio del sistema di controllo FBW, dei comandi primari di volo con attuazione idraulica, dell'addestratore militare di nuova generazione Aermacchi M346.

Il lavoro svolto propone una metodologia originale in risposta alle problematiche sopra esposte di individuazione delle avarie, gestione delle ridondanze e riconfigurazione del sistema.

In particolare si fa riferimento al caso concreto del sistema dati aria installato sulla parte prodiera della fusoliera del velivolo M346 e costituito da quattro sensori multi-funzione.

In letteratura al momento vengono descritte tecniche di gestione delle ridondanze relativamente alla determinazione della pressione a partire dai segnali forniti da quattro sonde ma il problema oggetto della presente tesi è in realtà molto più complesso. Infatti, come verrà spiegato in dettaglio nei paragrafi che seguono, per quanto concerne la determinazione degli angoli di incidenza e derapata, è indispensabile disporre contemporaneamente dei segnali forniti da due sensori differenti. Le stime disponibili sono dunque sei: si tratta quindi di una gestione non-standard delle ridondanze ed in merito la letteratura fornisce scarsissime informazioni. Inoltre poiché è possibile il verificarsi di avarie sia dei sensori angolari che di pressione, occorre prevedere diverse modalità operative in funzione delle possibili combinazioni di guasto.

A seguito di tali considerazioni, il contributo dato dalla presente tesi di dottorato può ritenersi del tutto originale. Nel lavoro viene descritto inoltre un esempio concreto di applicazione del *software Stateflow*<sup>®</sup>, per la realizzazione dei diversi modi operativi del sistema dati aria. Tale *toolbox* costituisce uno strumento molto potente di progettazione grafica di sistemi basati sulla teoria delle macchine a stati finiti (*Finite State Machine*, FSM), che consente di rappresentare un sistema complesso che evolve nel tempo in base agli eventi che si verificano. L'elemento qualificante di questo *software* risiede proprio nella possibilità di gestire e visualizzare gli stati attivi, ovvero le modalità operative del sistema, durante una generica simulazione. Inoltre, come verrà sottolineato nei paragrafi che seguono, il *software* consente di ridurre, rispetto ad una programmazione tradizionale, i tempi di calcolo poiché i modi non attivi non occupano risorse.

Il lavoro riporta inoltre uno studio approfondito dell'impatto delle possibili avarie dei sensori sulla stima dei parametri di volo, prendendo in considerazione, mediante un approccio alla Monte Carlo, le caratteristiche statistiche dei segnali forniti dalle sonde. Anche in presenza di un degrado delle prestazioni del sistema, legato al verificarsi di avarie di più sensori, gli errori commessi dalla procedura risultano ampiamente accettabili.

## 2.1 Il sistema dati aria di riferimento

Il sistema dati aria del velivolo Aermacchi M346 è caratterizzato dalla presenza di quattro sonde multi-funzione (*air data probes*), posizionate nella parte prodiera della fusoliera secondo la configurazione mostrata in Fig. 2.1.

Tale architettura è capace di sopperire alla presenza di un'avaria senza sensibili perdite di prestazioni ed operare in sicurezza anche quando il numero delle avarie è superiore ad uno (*fail operative-fail safe*).

Le sonde, prodotte dalla ditta americana ASI (*Avionics Specialities, Inc.*), sono della tipologia *IMFP*<sup>®</sup> *Integrated Multi Functional Probe*,<sup>[8][10]</sup> ed hanno la capacità di fornire misure affidabili in un ampio dominio ( $\alpha$ ,  $\beta$ ) grazie alla possibilità di allinearsi con la direzione locale del flusso nei punti di installazione (*null-seeking probes*).

La forma è tronco conica con asse normale alla superficie della fusoliera e sono dotate di cinque prese (*slot*) di pressione, di cui due vengono utilizzate per la rotazione intorno al proprio asse, vedi Fig. 2.2.

La generica sonda  $i$ -esima fornisce tre misure:  $\lambda_i$  angolo di flusso locale misurato da un apposito trasduttore angolare di posizione noto in letteratura come *Rotary Variable Differential Transducer* (RVDT), la *frontal pressure*  $P_{front_i}$  fornita dalla presa allineata alla direzione locale del flusso e la *slot pressure*  $P_{slot2_i}$  ottenuta come media delle pressioni in corrispondenza delle prese a  $90^\circ$  con la direzione del flusso.

Il pedice  $i$  ( $i=1, \dots, 4$ ) rappresenta l'indice della sonda.

## 2.2 Definizione del problema

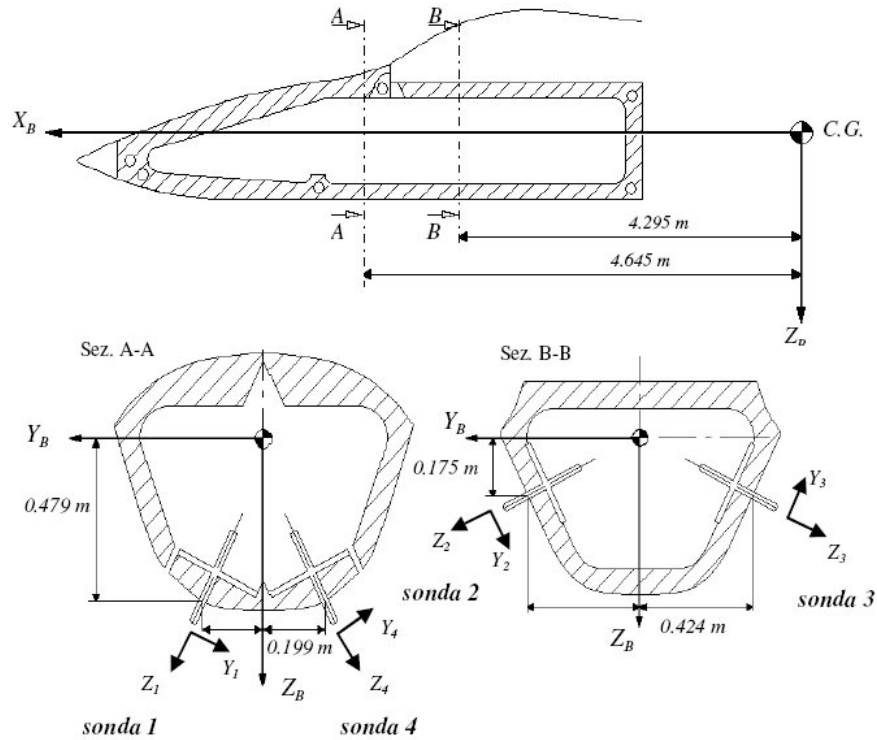
Come schematizzato in Fig. 2.3 la procedura di elaborazione dei dati aria è costituita da un insieme di algoritmi che hanno il compito di determinare i parametri di volo sulla base dei segnali forniti dalle sonde ( $\lambda_i$ ,  $P_{front_i}$ ,  $P_{slot2_i}$ ) ed assicurare una corretta gestione delle ridondanze dei sensori, indentificazione delle avarie e riconfigurazione del sistema.

La Fig. 2.3 evidenzia che gli algoritmi di elaborazione tengono anche conto degli effetti di manovra e della configurazione del velivolo (posizione dei carrelli, posizione dei flap, ...) in quanto le misure fornite dalle sonde sono influenzate dalla geometria del velivolo e dalla sua velocità angolare.

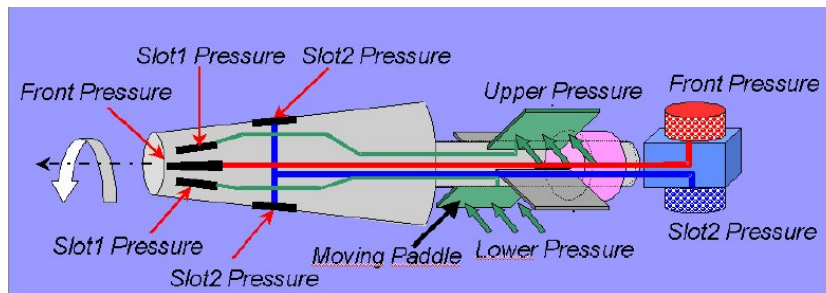
Dalla Fig. 2.3 si evince che parallelamente alla procedura di elaborazione è essenziale sviluppare dei modelli capaci di generare i segnali forniti dalle sonde, ovvero gli *input* della procedura di elaborazione stessa.

Tali modelli costituiscono la procedura di simulazione, ampiamente descritta in<sup>[11]</sup> : quest'ultima è stata utilizzata sia in fase di progettazione sia in fase





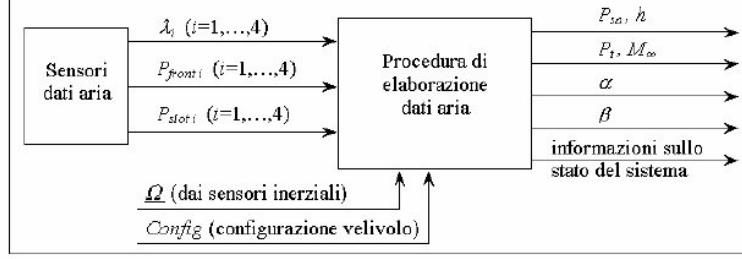
**Figura 2.1:** Posizione in fusoliera e sistemi di riferimento delle sonde dati aria.



**Figura 2.2:** Vista tridimensionale della sonda dati aria.

di simulazione su *Iron Bird* durante la quale viene verificata la funzionalità dell'intero *Flight Control System*.

La simulazione delle grandezze misurate dalle sonde *IMFP*<sup>®</sup> del sistema



**Figura 2.3:** Dati di ingresso ed uscite della procedura di elaborazione.

dati aria si basa sulla definizione delle seguenti funzioni:

$$\begin{cases} \lambda_i = f_{ai}(\alpha, \beta, M, Config, h) \\ Cps\_local_i = f_{bi}(\alpha, \beta, M, Config, h) \\ Cpt\_local_i = f_{ci}(\alpha, \beta, M, Config, h) \end{cases} \quad (2.1)$$

$$\begin{cases} \frac{P_{front_i}}{P_{t\_local_i}} = f_d \left( \frac{P_{t\_local_i}}{P_{s\_local_i}} \right) \\ \frac{P_{slot2_i}}{P_{s\_local_i}} = f_f \left( \frac{P_{t\_local_i}}{P_{s\_local_i}} \right) \end{cases} \quad (2.2)$$

$$\begin{cases} Cps\_local_i = \left( \frac{P_{s\_local_i} - P_{sa}}{\left(\frac{\gamma}{2}\right) \cdot P_{sa} \cdot M_{\infty}^2} \right) \\ Cpt\_local_i = \left( \frac{P_{t\_local_i} - P_{sa}}{\left(\frac{\gamma}{2}\right) \cdot P_{sa} \cdot M_{\infty}^2} \right) \end{cases} \quad (2.3)$$

Le Eq. (2.1) caratterizzano il campo aerodinamico sul *forebody* del velivolo (parte anteriore della fusoliera) correlando le condizioni di flusso locale nei punti di installazione degli *IMFP*<sup>®</sup> alle condizioni di flusso asintotico.

Le misure locali dei coefficienti di pressione  $Cps\_local_i$ ,  $Cpt\_local_i$  e dei  $\lambda_i$  sono funzioni degli angoli  $\alpha$ ,  $\beta$ , del numero di Mach  $M$ , della configurazione del velivolo e della quota  $h$  rispetto al suolo.

Gli angoli di flusso locali  $\lambda_i$  vengono determinati mediante misure dirette in galleria del vento su un modello del velivolo, in tutto l'involuppo di volo operativo, per diverse condizioni di volo, in moto rettilineo (velocità angolare  $\omega$  nulla).

Gli effetti di  $\omega$  vengono inseriti mediante un approccio alternativo descritto nei paragrafi successivi. Le Eq. (2.2) rappresentano le curve di taratura della sonda in condizioni *Stand Alone* e correlano le condizioni di flusso espresse

in termini di  $Pt\_local_i$  e  $Ps\_local_i$  alle misure  $P_{front_i}$  e  $P_{slot2_i}$ . Per ulteriori chiarimenti sul modello di sonda *Stand Alone* si rimanda a<sup>[12]</sup>.

A partire dalla definizione di coefficienti di pressione (Eq. (2.3)) e dalle Eq. (2.1) è possibile valutare i rapporti riportati nelle Eq. (2.4):

$$\begin{cases} \frac{Ps\_local_i}{P_{sa}} = \left(1 + \frac{\gamma}{2} \cdot Cps\_local_i \cdot M_\infty^2\right) \\ \frac{Pt\_local_i}{P_{sa}} = \left(1 + \frac{\gamma}{2} \cdot Cpt\_local_i \cdot M_\infty^2\right) \end{cases} \quad (2.4)$$

Combinando le Eq. (2.2) e le Eq. (2.4) si ottiene:

$$\begin{cases} \frac{P_{front_i}}{P_{sa}} = \frac{P_{front_i}}{Pt\_local_i} \cdot \frac{Pt\_local_i}{P_{sa}} = f_d \cdot \left(1 + \frac{\gamma}{2} \cdot Cpt\_local_i \cdot M_\infty^2\right) \\ \frac{P_{slot2_i}}{P_{sa}} = \frac{P_{slot2_i}}{Ps\_local_i} \cdot \frac{Ps\_local_i}{P_{sa}} = f_d \cdot \left(1 + \frac{\gamma}{2} \cdot Cps\_local_i \cdot M_\infty^2\right) \end{cases} \quad (2.5)$$

Nota dunque  $P_{sa}$  si moltiplicano per tale valore le Eq. (2.5) ottenendo in questo modo i valori simulati di  $P_{front_i}$  e  $P_{slot2_i}$ .

Si osservi che le Eq. (2.5) combinano le curve di taratura ( $f_d$  e  $f_f$ ) con le misure fatte in galleria del vento. Infatti la difficoltà di eseguire una campagna di prove in galleria del vento su un modello del velivolo comprensivo di un modello in scala delle sonde, ha comportato la necessità di determinare i coefficienti di pressione integrando le informazioni relative al campo aerodinamico intorno al velivolo nei punti di installazione delle sonde stesse, trascurando quindi parte degli effetti di interferenza.

## 2.3 Caratterizzazione del campo aerodinamico

La caratterizzazione del campo aerodinamico intorno al velivolo è essenziale per lo sviluppo delle due procedure sopracitate in quanto entrambe si basano sulla conoscenza delle relazioni che correlano i dati misurati con le condizioni asintotiche di volo.

Bisogna precisare però che nel presente lavoro gli effetti non stazionari sono stati trascurati poiché è complicato non solo realizzare prove in galleria del vento capaci di valutare tali effetti, ma anche sviluppare modelli teorici capaci di descriverli. Per tale motivo è stato deciso di trascurarli in questa prima fase del lavoro salvo, se necessario, reintrodurli in fase di confronto dei risultati con i dati delle prove di volo.

## 2.4 Effetti di manovra sulle funzioni aerodinamiche

Come già anticipato è difficile valutare le funzioni aerodinamiche in presenza della velocità angolare  $\omega$  del velivolo. Più ragionevole è pensare di realizzare

delle prove assimilabili a condizioni di moto rettilineo stazionario con velocità angolare nulla e tener conto degli effetti di manovra mediante lo sviluppo di modelli teorici basati su correzioni di tipo cinematico.

La presenza della velocità angolare  $\omega$  comporta una rotazione del velivolo intorno al baricentro con conseguente modifica del campo aerodinamico attorno ad esso. In particolare le componenti  $Q_B$  e  $R_B$  di  $\omega$  comportano una differenza tra il numero di Mach, gli angoli d'incidenza e derapata relativi al baricentro e quelli riferiti ad una generica sezione della fusoliera.

Questa differenza esistente tra valori baricentrici e di sezione porta a concludere che si può tener conto degli effetti di  $Q_B$  e  $R_B$  calcolando in condizioni stazionarie con  $\omega$  nulla le funzioni  $Cps\_local_i$ ,  $Cpt\_local_i$  e  $\lambda_i$  ed utilizzando i valori di sezione  $(\alpha_{si}, \beta_{si}, M_{si})$ , non quelli baricentrici  $(\alpha_B, \beta_B, M_B)$ . Si osservi che tali differenze non variano all'interno della sezione, cioè sono le stesse per tutte le sonde installate in quella sezione.

Il modello matematico usato esprime le dipendenze di  $\alpha_{si}$ ,  $\beta_{si}$ ,  $M_{si}$  dalle componenti della velocità baricentrica  $V_B$  e dalle componenti di velocità angolare  $Q_B$  e  $R_B$ .

Le componenti di velocità angolare introducono un campo di velocità aggiuntivo nelle sezioni d'installazione delle sonde dato da:

$$\begin{cases} u_{si} = u_B \\ v_{si} = v_B + X_{Bi} \cdot R_B \\ w_{si} = w_B - X_{Bi} \cdot Q_B \end{cases} \quad (2.6)$$

Inoltre essendo per definizione:

$$\begin{cases} V_{si} = \sqrt{u_{si}^2 + v_{si}^2 + w_{si}^2} \\ u_{si} = V_{si} \cdot \cos(\beta_{si}) \cdot \cos(\alpha_{si}) \\ v_{si} = V_{si} \cdot \sin(\beta_{si}) \\ w_{si} = V_{si} \cdot \cos(\beta_{si}) \cdot \sin(\alpha_{si}) \end{cases} \quad (2.7)$$

si ottiene:

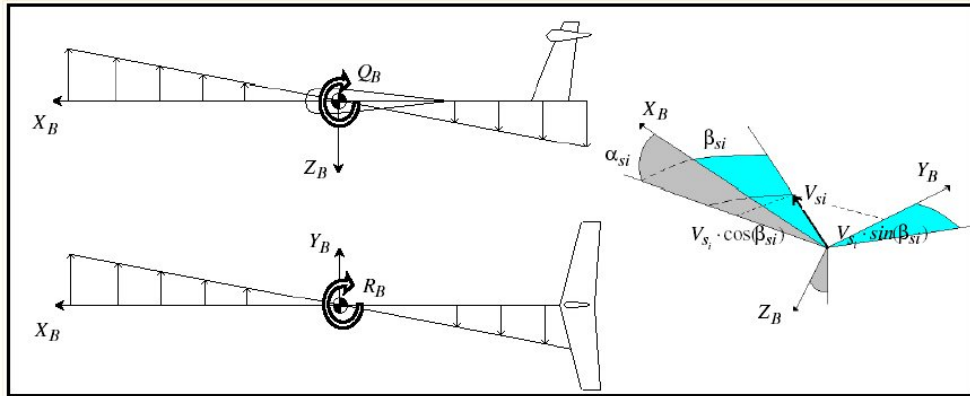
$$\begin{cases} \alpha_{si} = \arcsin\left(\frac{w_{si}}{u_{si}}\right) \\ \beta_{si} = \arctan\left(\frac{v_{si}}{\sqrt{u_{si}^2 + w_{si}^2}}\right) \\ M_{si} = \frac{\sqrt{u_{si}^2 + v_{si}^2 + w_{si}^2}}{a} \end{cases} \quad (2.8)$$

Si osservi dunque che i valori di sezione  $\alpha_{si}$ ,  $\beta_{si}$ ,  $M_{si}$  sostituiscono  $\alpha$ ,  $\beta$ ,  $M$  in Eq. (2.1) quando il velivolo si trova in fase di manovra.

Le componenti di velocità angolare  $Q_B$  e  $R_B$  introducono delle variazioni che, come già accennato, non dipendono dal punto di installazione delle sonde nella sezione considerata.

Al contrario le variazioni del campo di velocità introdotte da  $P_B$  dipendono dalla distanza dall'asse  $X_B$ : sono dunque diverse per le varie sonde e non sono state inserite nel presente modello di simulazione.

In Fig. 2.4 è mostrato uno schema del modello matematico qui introdotto.



**Figura 2.4:** Schematizzazione degli effetti delle velocità angolari  $Q_B$  e  $R_B$ .

## 2.5 Il database aerodinamico

Il *database* aerodinamico è stato ottenuto mediante *test* in galleria del vento su di un modello in scala del velivolo preso in esame. I *test* sono stati effettuati sia per la configurazione con superfici mobili deflesse e carrelli retratti (configurazione denominata di *Cruise*) sia per configurazioni con superfici mobili deflesse.

Le prove di galleria del vento sono state condotte per diversi valori del numero di Mach (circa 12 stazioni di Mach). Per ciascun valore, gli angoli di incidenza e di derapata sono stati fatti variare in un ampio dominio.

Il *database* sperimentale così ottenuto ed espresso sotto forma di *look-up table*, fornisce le grandezze ( $C_{ps\_local_i}$ ,  $C_{pt\_local_i}$  e  $\lambda_i$ ) per la configurazione di riferimento: l'effetto delle superfici di controllo del velivolo sulle suddette grandezze è espresso in termini di variazione rispetto a tale configurazione.

## 2.6 Avarie dei sensori dati aria

La procedura di simulazione, ampiamente descritta in<sup>[11]</sup>, riproduce i segnali misurati dalle sonde sia quando il sistema dati aria è completamente funzionante, sia quando presenta delle modalità d'avaria (*failure*). Per poter far ciò, è stato necessario sviluppare dei modelli capaci di simulare il comportamento delle sonde in queste condizioni (*lock*, *offset*, *departure*, ...). Tale studio ha portato all'identificazione di alcuni modi di avaria sia sulle misure angolari che su quelle di pressione, dettagliatamente descritte in<sup>[13]</sup>.

E' necessario sottolineare che nella procedura di simulazione realizzata, non è stata effettuata una modellizzazione fisica delle *failure* che è possibile inserire, ma è stato assunto un *set* di possibili forme d'onda da sovrapporre ai segnali simulati ed espandibile eventualmente in futuro.

Le forme d'onda da inserire sulle misure  $P_{front_i}$ ,  $P_{slot2_i}$  e  $\lambda_i$  sono di tipo sinusoidale, gaussiana, a rampa, *offset*, *freeze* ed è possibile sia sovrapporre le forme d'onda sia intervenire di volta in volta su ampiezza, fase, istante d'inserimento, ... .

## 2.7 Il software di simulazione

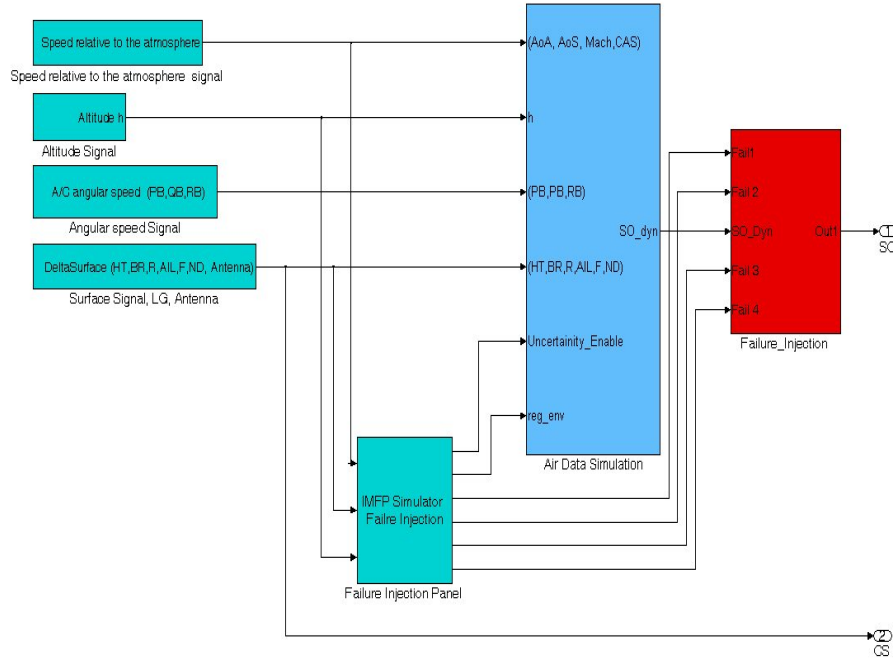
La procedura di simulazione è costituita da un insieme di modelli che permettono di riprodurre le grandezze misurate dalle sonde del sistema dati aria ( $P_{front_i}$ ,  $P_{slot2_i}$ ,  $\lambda_i$ ).

Bisogna precisare che l'algoritmo sviluppato presenta diverse aree funzionali che permettono di effettuare diverse tipologie di simulazione:

- Modello *Nominal*: consente la simulazione dei segnali in condizioni ideali;
- Modello *Tolerance*: consente di effettuare una simulazione più realistica tramite l'inserimento di tolleranze aerodinamiche (si tratta di incertezze sui segnali ideali, fornite dal costruttore);
- *Failure Injection*: consente di inserire e sovrapporre i modi d'avaria precedentemente introdotti.

La procedura di simulazione è stata realizzata in ambiente *Matlab/Simulink*<sup>®</sup> ed in Fig. 2.5 ne viene mostrato il *level 0*.

Il blocco "*Speed relative to the atmosphere signal*" fornisce alla procedura di simulazione gli angoli d'attacco AoA, di *sideslip* AoS, il numero di Mach ed il valore della *calibrated air speed*, CAS: tali ingressi possono assumere valore costante o variabile durante la simulazione.



**Figura 2.5:** *Level 0* della procedura di simulazione dati aria.

Il blocco "*Altitude signal*" permette di determinare la *pressure altitude* o quota velivolo riferita al *sea level* e la *ground height* o quota velivolo riferita al suolo.

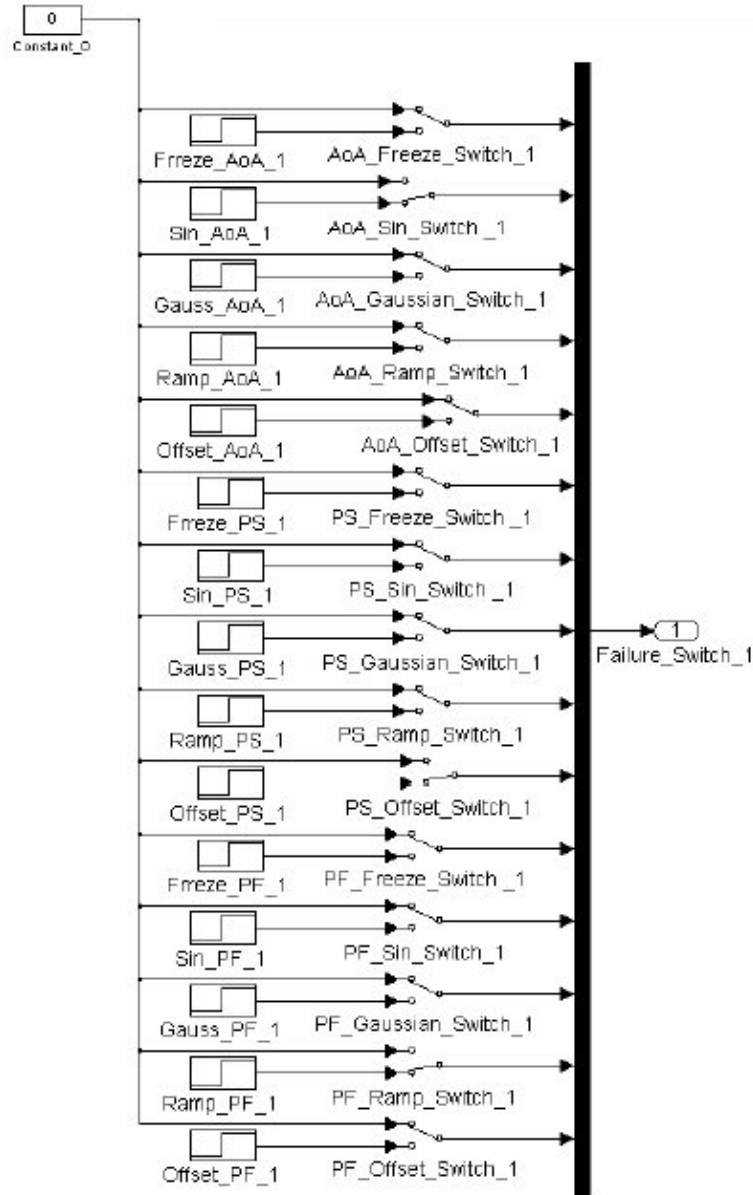
Tramite il blocco "*Angular speed signal*" è possibile inserire, in radianti, le componenti  $(P_B, Q_B, R_B)$  della velocità angolare: anche in questo caso gli ingressi possono essere costanti o variabili, in funzione delle caratteristiche della manovra effettuata.

Il modello consente l'inserimento delle deflessioni delle superfici mobili *flap* F, *nose droop* ND, la posizione del carrello LG, dell'IRFP e dell'antenna: si osservi che quest'ultimi due contributi sono sempre presenti.

Il blocco "*Air Data Simulation*" fornisce i valori dei parametri  $P_{front_i}$ ,  $P_{slot2_i}$ ,  $\lambda_i$  in condizioni ideali ovvero *failure free*.

In "*Failure Injection Panel*" è possibile inserire sia le tolleranze aerodinamiche sui  $Cps_{local_i}$ ,  $Cpt_{local_i}$  e  $\lambda_i$ , sia le principali modalità di avaria, controllando tale inserimento attraverso un opportuno pannello (in Fig. 2.6 è mostrato in particolare quello relativo alla sonda 1). Il modello simula anche otto variabili booleane che hanno il compito di descrivere lo stato di avaria delle misure associate alla sonda  $i$ -esima (parte a destra in Fig. 2.7).

La sovrapposizione delle tolleranze e delle avarie sui segnali *failure free* av-

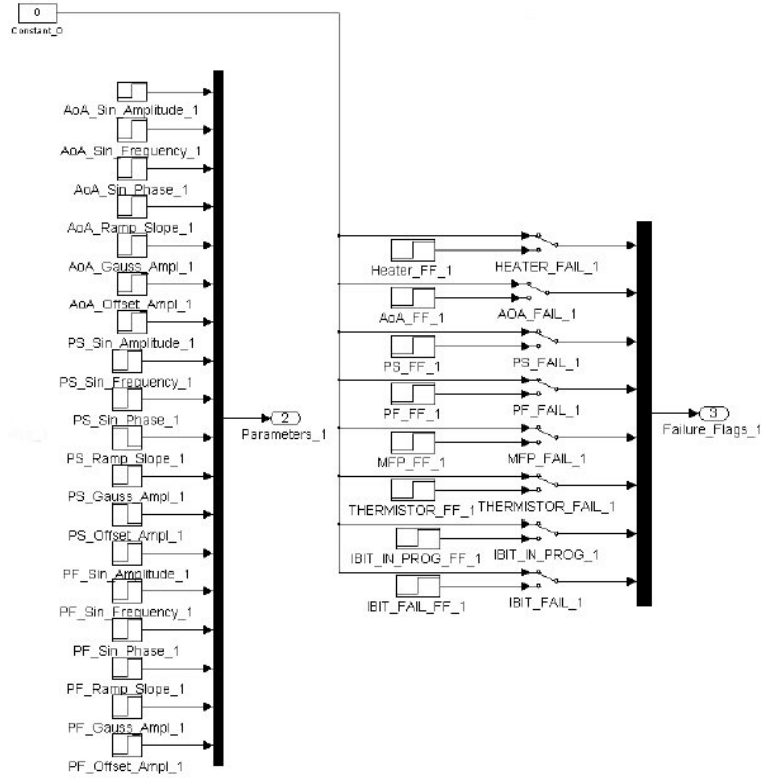


**Figura 2.6:** Pannello di inserimento delle avarie sulla sonda 1.

viene nel blocco "*Failure Injection*".

Per una descrizione dettagliata della modello di simulazione e di tutti i bloc-





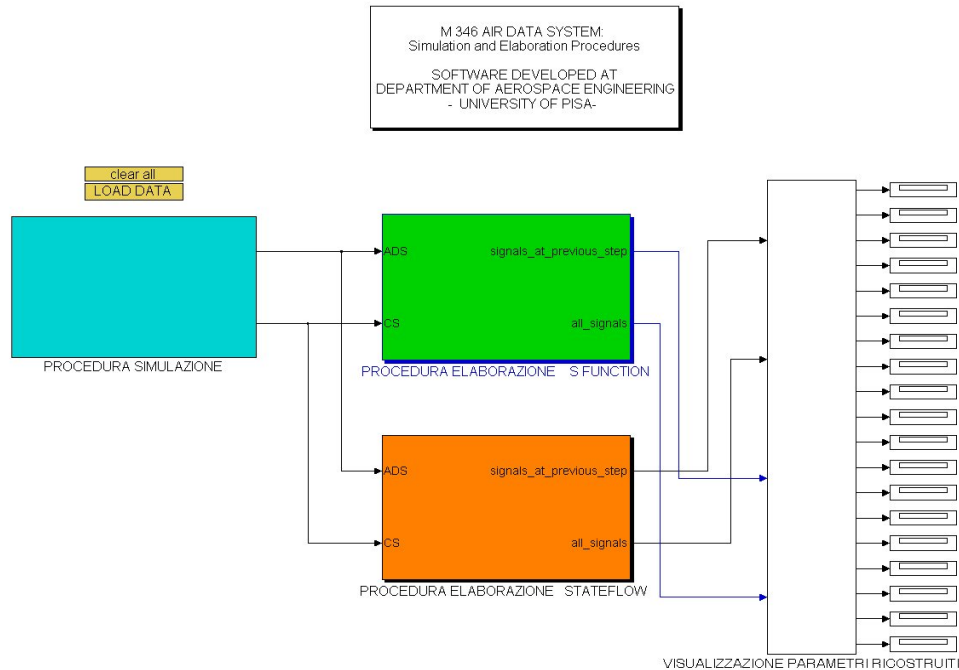
**Figura 2.7:** Pannello di inserimento delle avarie sulla sonda 1.

chi *Simulink*<sup>®</sup> che lo costituiscono si rimanda a<sup>[11]</sup>.

## 2.8 La procedura di elaborazione

La Fig. 2.8 mostra il *level 0* del sistema dati aria del velivolo M346 realizzata in *Matlab/Simulink*<sup>®</sup>. E' possibile osservare il blocco relativo alla procedura di simulazione, descritto in precedenza, e quelli relativi alla procedura di elaborazione che come anticipato ha il compito di determinare le incognite ( $\alpha$ ,  $\beta$ ,  $P_{sa}$ ,  $P_t$ ) note le grandezze misurate dalle sonde ( $P_{front_i}$ ,  $P_{slot2_i}$ ,  $\lambda_i$ ).

Come si può notare, la procedura di elaborazione è stata realizzata sia tramite una programmazione tradizionale in *C++* sia utilizzando il *software Stateflow*<sup>®</sup>: nel seguito del presente capitolo verrà effettuata una descrizione delle logiche di gestione delle ridondanze del sistema dati aria, con particolare attenzione alla loro realizzazione proprio in ambiente *Stateflow*.



**Figura 2.8:** Level 0 del sistema dati aria.

Il blocco "*Visualizzazione parametri ricostruiti*" raccoglie i segnali d'uscita tramite blocchetti *mux* e verifica che entrambe le programmazioni forniscano gli stessi risultati, visualizzandoli con degli *scope*.

### 2.8.1 Gestione delle ridondanze

L'algoritmo di elaborazione dei dati aria deve curare oltre alla fase computazionale anche gli aspetti legati alla gestione delle ridondanze del sistema. L'algoritmo effettua diverse stime delle grandezze derivate: tali stime però sono solo teoricamente coincidenti, in pratica ciascuna è affetta da errori differenti, imputabili all'accuratezza delle sonde o ad approssimazioni dell'algoritmo di elaborazione stesso.

A volte le stime possono esser discordi tra loro a causa di condizioni di avaria. Sono state individuate e prese in considerazione due tipi di avarie: quelle che producono la perdita di misure angolari (ad esempio la perdita di un trasduttore angolare) e quelle che comportano la perdita di misure di pressione (perdita di un trasduttore di pressione). Non è da escludere però il verificarsi di avarie che comportano la perdita contemporanea di entrambe le misure.

E' importante dunque la scelta della tipologia e dei punti di installazione delle sonde in modo da garantire un sistema operativo per tutti gli assetti ( $\alpha$  -  $\beta$ ) ammissibili nell'involuppo di volo considerato.

Per i velivoli ad alte prestazioni, poiché il campo di variazione degli assetti è estremamente ampio, può essere difficile assicurare che non esistano condizioni in cui una sonda si venga a trovare nella scia di altre parti del velivolo. Alla luce di queste considerazioni, si pongono due problematiche.

L'algoritmo deve fornire al mondo esterno (leggi di controllo, sistemi avionici, ...) un unico valore per ogni parametro derivato e quindi necessita di algoritmi di *voting*.

Tuttavia tali algoritmi devono operare solo su stime provenienti da sonde che non presentano condizioni di avaria. Questo aspetto richiede la presenza di algoritmi di *monitoring*, ovvero funzioni logiche capaci di rilevare continuamente ed isolare le avarie delle sonde ma allo stesso tempo insensibili agli errori sistematici legati al processo di calcolo (errori di approssimazione delle funzioni, accuratezza delle sonde, ...).

Quest'ultimi vanno ad affiancare il *Built in Test* (BIT) ovvero un *monitoring* realizzato direttamente dalle sonde stesse che ha il compito di associare un segnale di validità ad ogni misura da esse fornita.

Nel presente lavoro sono stati utilizzati gli algoritmi di *cross channel monitoring* che confrontano i segnali provenienti dalle quattro linee indipendenti del processo di calcolo e controllano le differenze esistenti tra questi segnali, verificando se sono superiori ad opportune soglie di accettazione.

E' importante sottolineare che la taratura di tali soglie gioca un ruolo centrale, poiché una calibrazione non corretta potrebbe provocare la dichiarazione di avarie inesistenti o apparenti. Infatti da una parte è pericoloso non individuare le avarie delle sonde, ma è tuttavia da evitare che errori sistematici vengano interpretati come avarie delle sonde.

L'architettura del sistema dati ariia oggetto del presente lavoro è caratterizzata da una ridondanza di tipo quadruplex, capace dunque di sofferire alla presenza di una *failure* senza sensibili perdite di prestazioni e di operare comunque in sicurezza quando il numero è maggiore di uno.

Il sistema dati ariia si comporta dunque come un sistema a triplice ridondanza relativamente alla ricostruzione degli angoli (*fail operative* - *fail safe*) mentre è garantita una quadruplice ridondanza per il processo di calcolo delle pressioni (*two fail operative* - *fail safe*).

La possibilità di separare il problema della ricostruzione degli angoli da quello delle pressioni, non discussa in questo lavoro, da luogo a due fasi di *monitoring* ed altrettante di *voting*.

### 2.8.2 Algoritmi di monitoring angoli

Per quanto riguarda la ricostruzione degli angoli di incidenza e di derapata, l'algoritmo ha a disposizione sei possibili coppie  $(\lambda_i, \lambda_j)$  non corrispondenti però ad una sestupla ridondanza.

La Tabella 2.1 mostra come varia il numero delle coppie disponibili al verificarsi delle avarie delle sonde. La perdita della generica misura  $\lambda_i$  relativa alla direzione di flusso locale, infatti, comporta che le coppie disponibili diventino solo tre, ancora sufficienti tuttavia per generare valori consolidati di  $\alpha$  e  $\beta$ .

Alla seconda avaria angolare, le rimanenti due sonde funzionanti danno luogo ad un'unica coppia  $(\lambda_i, \lambda_j)$  dalla quale può essere ottenuto un unico valore di  $\alpha$  ed un unico valore di  $\beta$ . Tali valori non possono essere sottoposti ad un processo di *monitoring* in quanto mancano valori di confronto e quindi non è possibile fornire al modo esterno valori consolidati.

In definitiva il processo di calcolo degli angoli può sopperire ad una sola avaria riguardante le misure angolari e per tale ragione si ha un comportamento assimilabile a quello di un sistema triplex.

Gli algoritmi di *monitoring* realizzati dalla procedura di elaborazione sono,

coppie disponibili	avaria sonda 1	avaria sonda 2	avaria sonda 3	avaria sonda 4
<i>coppia</i> (1, 2)			X	X
<i>coppia</i> (1, 3)		X		X
<i>coppia</i> (1, 4)		X	X	
<i>coppia</i> (2, 3)	X			X
<i>coppia</i> (2, 4)	X		X	
<i>coppia</i> (3, 4)	X	X		

**Tabella 2.1:** Coppie disponibili al variare delle sonde in avaria.

come detto in precedenza, di tipo *cross-channel monitoring* poiché individuano le avarie mediante confronti tra i vari canali di ridondanza.

In particolare gli algoritmi relativi alla ricostruzione degli angoli operano dei confronti tra le grandezze  $\alpha$  e  $\beta$  derivate dalle misure delle sonde. Per riconoscere la prima avaria angoli, la procedura forma quattro gruppi, ciascuno dei quali è costituito da tre stime di  $\alpha$  derivanti da tre coppie  $(\lambda_i, \lambda_j)$ .

Ogni gruppo è identificato da un numero, da 1 a 4, che indica l'unica sonda non utilizzata per formare le tre coppie  $(\lambda_i, \lambda_j)$  in esame.

Seguendo questo schema, i quattro gruppi sono costituiti dai valori riportati in Tabella 2.2.

E' importante sottolineare che ad ogni *frame* di esecuzione, la procedura

gruppo 1	gruppo 2	gruppo 3	gruppo 4
$\alpha_{23}, \alpha_{24}, \alpha_{34}$	$\alpha_{13}, \alpha_{14}, \alpha_{34}$	$\alpha_{12}, \alpha_{14}, \alpha_{24}$	$\alpha_{12}, \alpha_{13}, \alpha_{23}$

**Tabella 2.2:** Valori degli angoli costituenti i gruppi.

ordina i tre valori appartenenti ad ogni gruppo e verifica che gli scostamenti tra i due valori estremi (il più piccolo ed il più grande) e quello centrale siano inferiori ad un preciso valore che rappresenta la soglia di *monitoring*,  $th\_alfa$ . L'opportuna taratura di questa soglia permette di risalire alla sonda in avaria tramite le stime di  $\alpha$ , in quanto nel momento in cui la sonda  $i$ -esima presenta una *failure*, in tutti i gruppi sarà rilevato il superamento della soglia tranne in quello in cui la sonda non è presente.

La Fig. 2.9 riporta un esempio di riconoscimento di avaria mediante la logica descritta: nel caso b) la sonda 1 si trova in avaria mentre nel caso a) non sono presenti avarie.

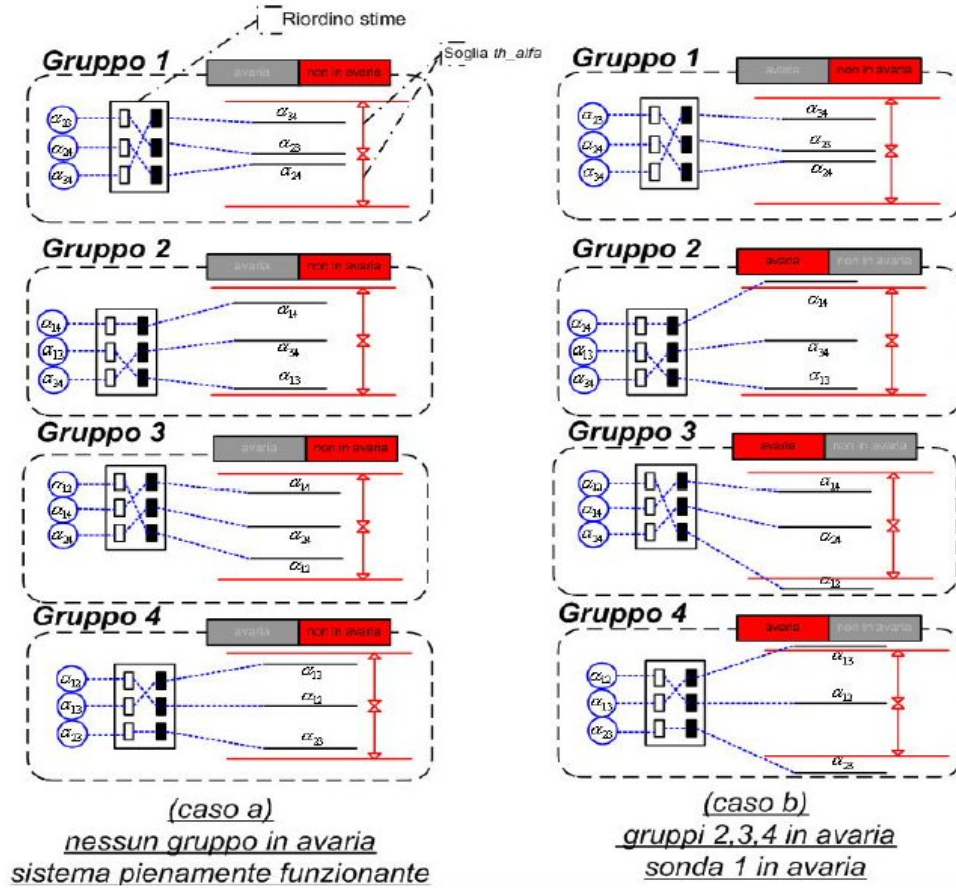
Se un'avaria è già stata individuata, si hanno solo tre coppie di angoli ( $\lambda_i, \lambda_j$ ). In tal caso ordinati i tre valori di  $\alpha$  associati a queste tre coppie il superamento della soglia di *monitoring* dello scostamento tra il valore più grande e quello centrale e/o tra il più piccolo e quello centrale determina la dichiarazione di una seconda avaria, anche se come detto in precedenza non è possibile individuare tale sonda.

Per quanto riguarda la stima dell'angolo di *sideslip*  $\beta$ , nella procedura di elaborazione è presente una logica di *monitoring* del tutto analoga a quella descritta per  $\alpha$ .

Nel momento in cui almeno uno dei due algoritmi di *monitoring* individua l'avaria di una sonda, la procedura di elaborazione esclude dal processo di calcolo le misure ad essa associate.

### 2.8.3 Algoritmi di monitoring pressioni

Relativamente alla ricostruzione delle pressioni, il sistema ha una maggiore capacità di sopperire alla presenza di avarie in quanto le quattro sonde garantiscono una quadruplice ridondanza, ovvero ogni sonda fornisce una stima indipendente della pressione statica  $P_{sa}$ , della pressione totale  $P_t$ . Gli algoritmi di *monitoring* sono quindi attivi anche in presenza di due sonde in avaria. Come per gli angoli, una possibile metodologia per ipotizzare la presenza di una *failure*, è quella di verificare se uno degli quattro parametri elaborati  $P_{sa_i}$  (analogamente per le quattro  $P_{t_i}$ ) si discosta dal valore reale della pressione di una quantità superiore ad una opportuna soglia di accettazione



**Figura 2.9:** Logica di funzionamento dell'algoritmo di *monitoring* angoli.

( $th\_Psa$  per la pressione statica e  $th\_Pt$  per la pressione totale).

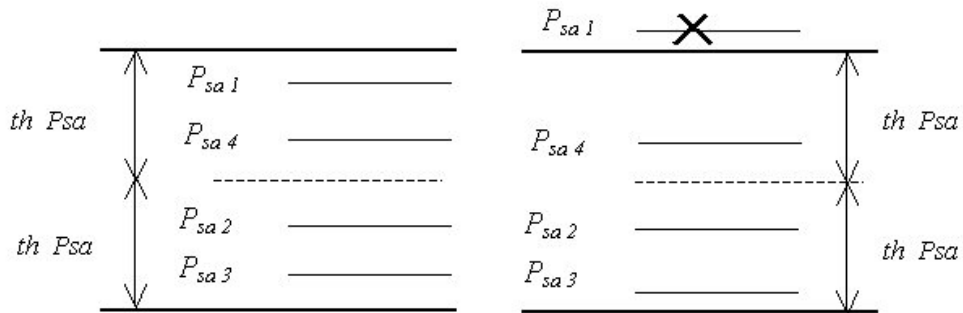
poiché i valori reali delle pressioni non sono noti, il confronto verrà effettuato con le loro migliori approssimazioni,  $Psa_m$  e  $Pt_m$ . Tali valori vengono calcolati in modo diverso a seconda del numero di sonde già dichiarate in *failure* al momento in cui viene eseguito il *monitoring*:

- Nessuna avaria dichiarata: dopo aver ordinato i quattro valori  $Psa_i$ ,  $Psa_m$  viene calcolato come media dei due valori centrali;
- Un'avaria dichiarata: dopo aver ordinato i tre valori  $Psa_i$ ,  $Psa_m$  coincide col valore centrale;
- Due avarie dichiarate: nel caso in cui le sonde in avarie sono due, la

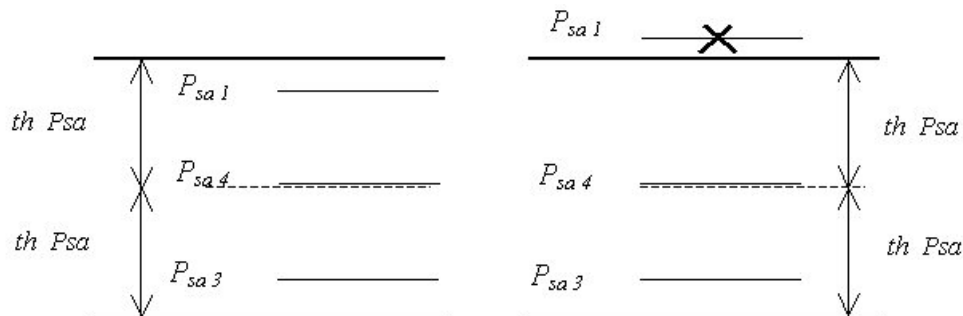
procedura verifica che lo scostamento tra i due unici valori disponibili (relativi alle sonde non affette da *failure*) sia inferiore a  $th\_Psa$ . Se tale controllo non è superato, la procedura dichiara l'esistenza delle terza avaria ma non riesce a stabilire quale sonda ne risulta affetta.

Analogamente per quanto riguarda i valore di confronto  $Pt_m$ .

Le figure che seguono mostrano graficamente l'individuazione di una avaria relativa alla sonda 1 da parte dell'algoritmo di *monitoring* sulla pressione statica. Dapprima si esamina una condizione iniziale dichiarata di pieno funzionamento e successivamente viene mostrata l'individuazione di una sonda in avaria (sonda 1) questa volta a partire da soli tre segnali  $Psa_i$  buoni (sonda 2 in avaria dichiarata).



**Figura 2.10:** Algoritmo di *monitoring* sulle pressioni: individuazione di una sonda in avaria partendo da condizioni di pieno funzionamento.



**Figura 2.11:** Algoritmo di *monitoring* sulle pressioni: individuazione di una sonda in avaria partendo da un'avaria dichiarata.

### 2.8.4 Consolidamento delle avarie

La presenza di disturbi esterni a cui il sistema può esser soggetto (turbolenza, raffiche, ...), rende necessario che un'avaria, dopo esser stata riconosciuta per la prima volta, sia sottoposta ad una fase di consolidamento per verificare la persistenza dell'avaria corrente.

Infatti la rilevazione di eventuali condizioni di avaria del sensore non deve sempre determinare l'esclusione definitiva di quest'ultimo dai processi di calcolo della procedura di elaborazione dati aria.

Un'avaria sarà dichiarata irreversibile e, quindi, considerata consolidata solo se l'algoritmo di *monitoring* la rileva un dato numero di volte ( $n\_frame$ ) consecutive, corrispondente ad un preciso intervallo  $\Delta t$ . Quest'intervallo di tempo deve esser scelto opportunamente maggiore rispetto ai tempi di estinzione dei disturbi, ma anche tale da non compromettere la stabilità del velivolo.

Se l'avaria viene rilevata un numero di volte inferiore a  $n\_frame$ , la sonda interessata non deve esser esclusa dai processi di calcolo eseguiti dalla procedura nei *frame* successivi.

### 2.8.5 Il problema del Low Speed

Gli algoritmi di *monitoring* e di *voting* possono entrare in crisi in una particolare condizione di volo.

Le sonde riescono a garantire misure affidabili fino a quando il flusso che le investe è caratterizzato da una pressione dinamica tale da consentirne l'autoallineamento alla direzione locale del flusso. Esiste quindi una velocità del velivolo al di sotto della quale le misure fornite dalle sonde non sono più affidabili. Questa condizione del sistema è denominata *Low Speed* e durante una missione si può manifestare in più occasioni (descrizione dettagliata in<sup>[14]</sup> e<sup>[15]</sup>). Ad esempio se si pensa alla fase del decollo il velivolo avrà sicuramente per un dato periodo una velocità inferiore a quella di minima validità dei segnali del sistema dati aria.

Durante il *Low Speed* è necessario che il *monitoring* sia parzialmente inibito e limitato all'analisi fatta dal BIT, poiché la presenza di segnali poco affidabili si traduce in errori considerevoli sulle grandezze ricostruite. Questi errori potrebbero esser interpretati dall'algoritmo di *monitoring* come avarie non reali.

Per sospendere la fase di *monitoring* è fondamentale il riconoscimento della condizione di *Low Speed* da parte della procedura, sfruttando le informazioni provenienti dalle sonde stesse: nel FCS la velocità è ricavata dalle grandezze fornite dal solo sistema dati aria. Quando il velivolo ha una velocità inferiore a quella di minima validità dei segnali dati aria, la pressione dinamica che



investe la sonda non è sufficiente ad instaurare una differenza di pressione  $\Delta P$  tale da consentire alla sonda stessa di allinearsi alla direzione locale del flusso. Conseguenza di ciò è che il valore del rapporto  $P_{front_i}$  su  $P_{slot2_i}$  è inferiore a quello aspettato: infatti in questa condizione la pressione  $P_{front_i}$  è minore di quella che si avrebbe nelle stesse condizioni di velocità se la sonda riuscisse ad allinearsi alla direzione locale del flusso. Al contrario la  $P_{slot2_i}$  non dovrebbe presentare sensibili differenze.

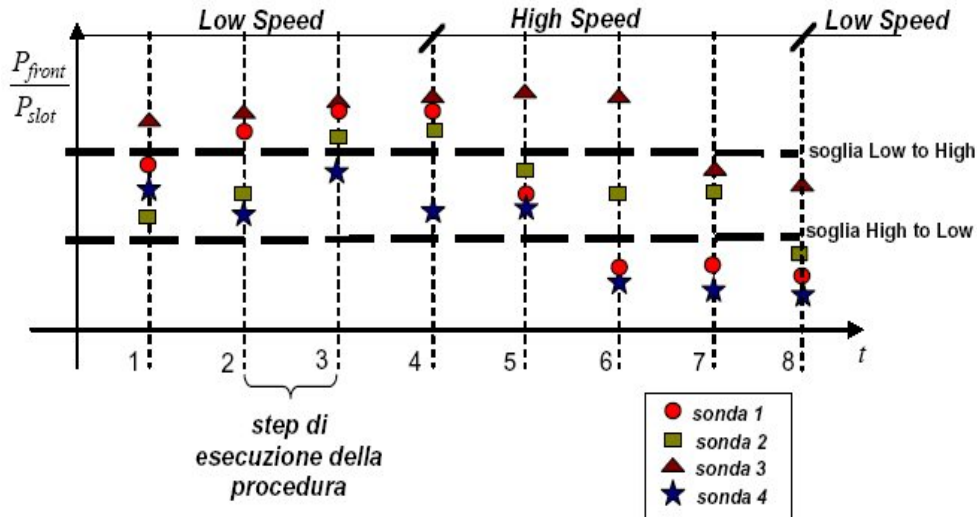
Un modo per identificare la condizione *Low Speed* è quindi quello di controllare il rapporto delle pressioni misurate dalle sonde, funzione del numero di Mach. Si osservi che il modello è stato sviluppato in modo da garantire che la procedura esca sempre dal *Low Speed* in fase di decollo. In particolare per verificare se il velivolo si trova in condizione di *Low Speed* il rapporto  $P_{front_i}/P_{slot2_i}$  può esser confrontato con un'opportuna soglia determinata inizialmente sulla base dei dati provenienti da galleria del vento e successivamente calibrata con le prove di volo.

Il verificarsi di disturbi esterni (ad esempio fenomeni di turbolenza, raffiche, ...) oppure la presenza di rumore sui segnali forniti dalle sonde potrebbero provocare oscillazioni dei rapporti  $P_{front_i}/P_{slot2_i}$  intorno alla soglia. Per evitare questo tipo di problema si è deciso di considerare due differenti soglie, una per determinare l'uscita dal *Low Speed*, rappresentativa del passaggio da una condizione di bassa ad una di alta velocità (*Low to High*); un'altra rappresentativa invece del passaggio da alta a bassa velocità (*High to Low*). Si osservi che la soglia *Low to High* deve esser maggiore di *High to Low* altrimenti il sistema una volta uscito dalla condizione di *Low* potrebbe non rientrarvi più.

In realtà per realizzare il passaggio da uno stato del sistema all'altro non è sufficiente che due sonde verifichino la condizione di passaggio: si richiede anche che tale condizione sia confermata dalle due sonde per un opportuno numero di *step* di esecuzione consecutivi ( $n_{frame}$ ).

La Fig. 2.12 mostra il passaggio *Low to High* ed *High to Low* (a titolo d'esempio si è supposto per semplicità che  $n_{frame}$  sia pari a tre). Al primo *frame* d'esecuzione il sistema si trova in una condizione di *Low Speed* e quindi la procedura controlla il passaggio dalla condizione di *Low* a quella di *High Speed*. A tale *frame* solo il rapporto di pressione della sonda 3 supera la soglia *Low to High* mentre dal secondo in poi anche la sonda 1 verifica la condizione. Al quarto *frame*, il sistema dichiara la condizione di *High Speed* poiché le sonde 1 e 3 hanno verificato la condizione di passaggio *Low to High* per un numero di *step* pari a  $n_{frame}$ . Al *frame* sei, sia la sonda 1 che la sonda 4 sono al di sotto della soglia *High to Low* e rimangono in tale condizione anche nei due *step* successivi in modo che allo *step* 8 la procedura può dichiarare la

condizione di *Low Speed*.



**Figura 2.12:** Visualizzazione del passaggio dalla condizione di *Low Speed* a quella di *High Speed*.

### 2.8.6 Algoritmi di voting

Come detto in precedenza, la procedura deve fornire al mondo esterno (leggi di controllo, sistemi avionici, ...) un unico valore per ogni parametro derivato e quindi necessita di algoritmi capaci di generare tali valori a partire dalle diverse stime realizzate.

Gli algoritmi della procedura predisposti a svolgere questo compito sono quelli di *voting*: l'esigenza di fornire all'esterno dei valori non affetti da errori dovuti al verificarsi di avarie, comporta la necessità che gli algoritmi di *voting* interagiscano con quelli di *monitoring* e con il BIT operato dalle sonde.

Il processo di *voting* quindi sarà dipendente dal numero di sonde in avaria rilevate dal sistema.

In condizione di piena funzionalità del sistema, il *voting* ordina in senso crescente tutte le stime disponibili (sei per la ricostruzione di  $\alpha$ , sei per la ricostruzione di  $\beta$ , quattro per la ricostruzione di  $P_{sa}$  e altrettante per  $P_t$ ) e prende la media dei due valori centrali, poiché s'ipotizza che essi siano i valori che meglio stimano le grandezze da ricostruire.

Alla presenza di un'avaria angoli o pressioni, gli algoritmi di *voting* hanno a disposizione solo tre stime per ciascuna delle grandezze ricostruite.

In tale situazione il parametro votato è il valore che occupa la posizione cen-

trale per ciascuna tripletta ordinata in senso crescente.

Quando sono presenti due avarie di angoli, la procedura di elaborazione, come già detto, non è in grado di fornire stime consolidate e di conseguenza gli algoritmi di *voting* angoli non sono attivi.

Nel caso invece di due avarie di pressione, è ancora possibile fornire stime consolidate di pressione statica e pressione totale in quanto la procedura di elaborazione consente una quadruplica ridondanza relativamente a tale processo di calcolo. In questa ultima condizione, i valori votati sono ottenuti eseguendo la media delle due stime disponibili di  $P_{sa}$  e delle due stime di  $P_t$ .

numero avarie	Ricostruzione angoli	
	numero stime	valore votato
nessuna	6	media centrali
una	3	valore centrale
due	1	nessuno
tre	nessuna	nessuno

**Tabella 2.3:** Algoritmi di *voting* angoli.

numero avarie	Ricostruzione pressioni	
	numero stime	valore votato
nessuna	4	media centrali
una	3	valore centrale
due	2	media
tre	nessuna	nessuno

**Tabella 2.4:** Algoritmi di *voting* pressioni.

### 2.8.7 Modi operativi della procedura di elaborazione dati aria

Alla luce di quanto detto in precedenza, la molteplicità di stime legate alla ridondanza dei sensori consente di riconfigurare il sistema in presenza di avarie. Il verificarsi infatti di una o più avarie causa il degrado delle prestazioni del sistema o addirittura l'impossibilità di derivare i parametri di volo.

Ciò porta alla necessità di definire diversi modi operativi di funzionamento della procedura, ognuno caratterizzato da un numero diverso di processi.

L'attivazione del singolo modo è legata al tipo di avaria rilevata dagli algoritmi di *monitoring*.

La procedura di elaborazione dei dati aerea, oggetto del presente lavoro, prevede i seguenti modi operativi:

- *Normal Mode*: rappresentativo della piena funzionalità del sistema e capace di fornire i valori consolidati di tutti i parametri di volo;
- *Flow Fail Mode*: caratterizzato dai soli processi di calcolo delle pressioni;
- *By Pass Mode*: rappresentativo della condizione in cui tutti i processi di calcolo sono inibiti e nessun parametro è fornito all'esterno;
- *Low Speed Mode*: l'attivazione di questo modo avviene quando il sistema individua la *low speed condition* (velocità del velivolo al di sotto dei 50kts). I parametri di volo vengono settati in modo da rispettare i requisiti di questa particolare condizione di volo;
- *Freeze Mode*: quando il velivolo si trova nella *out of envelope condition*, ovvero fuori dall'involucro di volo ( $\alpha$ ,  $\beta$ ) considerato, i parametri che vengono forniti al mondo esterno sono quelli calcolati allo *step* precedente all'ingresso nel *freeze mode*.

La Fig. 2.13 evidenzia le transizioni tra i diversi modi operativi prima elencati, funzioni del numero di avarie e della condizione di *Low Speed*.

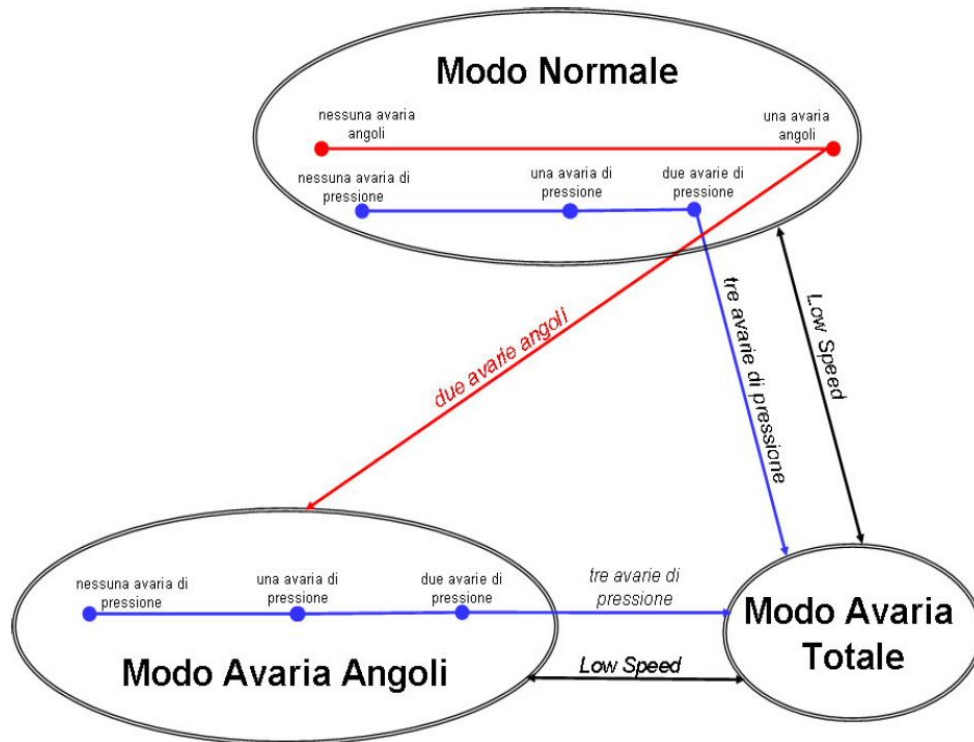
Si osservi che l'uscita dal *Normal Mode* è possibile quando :

- il velivolo è in condizione di *Low Speed*;
- la procedura ha identificato due avarie relative agli angoli;
- la procedura ha identificato tre avarie relative alle pressioni.

Il passaggio dal *Normal* al *Flow Fail Mode* avviene sia in seguito all'identificazione della terza avaria di pressione sia in seguito al riconoscimento della condizione di *Low Speed*.

La seconda avaria sugli angoli comporta invece la transizione dal *Normal* al *By Pass Mode*.

E' importante osservare che mentre il verificarsi di una o più avarie comporta sempre delle transizioni irreversibili, il verificarsi della condizione di *Low Speed* provoca invece il passaggio reversibile al *By Pass Mode* in quanto al decadere di tale condizione il sistema può tornare nella modalità di funzionamento precedente.



**Figura 2.13:** Modi operativi della procedura di elaborazione dati ari.

La Fig. 2.14 mostra la realizzazione in ambiente *Matlab/Simulink*<sup>®</sup> dei diversi modi operativi descritti in precedenza.

Anche se nel presente lavoro non verrà effettuata una descrizione dettagliata di tutte le logiche che costituiscono la procedura di elaborazione, è interessante osservare che il blocchetto "*External Capability*", in funzione degli ingressi provenienti dalla procedura di simulazione *ADS* e delle superfici di controllo *CS*, valuta se il velivolo si trova in condizione di *low speed* o *out of envelope* ed aggiorna opportune variabili *booleane* denominate *flag*.

In base al valore assunto da quest'ultime, la procedura provvederà ad attivare uno dei modi operativi rappresentati dai blocchetti successivi.

I segnali così ottenuti verranno successivamente ordinati nel "*Subsystem3*", per poi entrare in "*Visualizzazione parametri ricostruiti*" di Fig. 2.8.

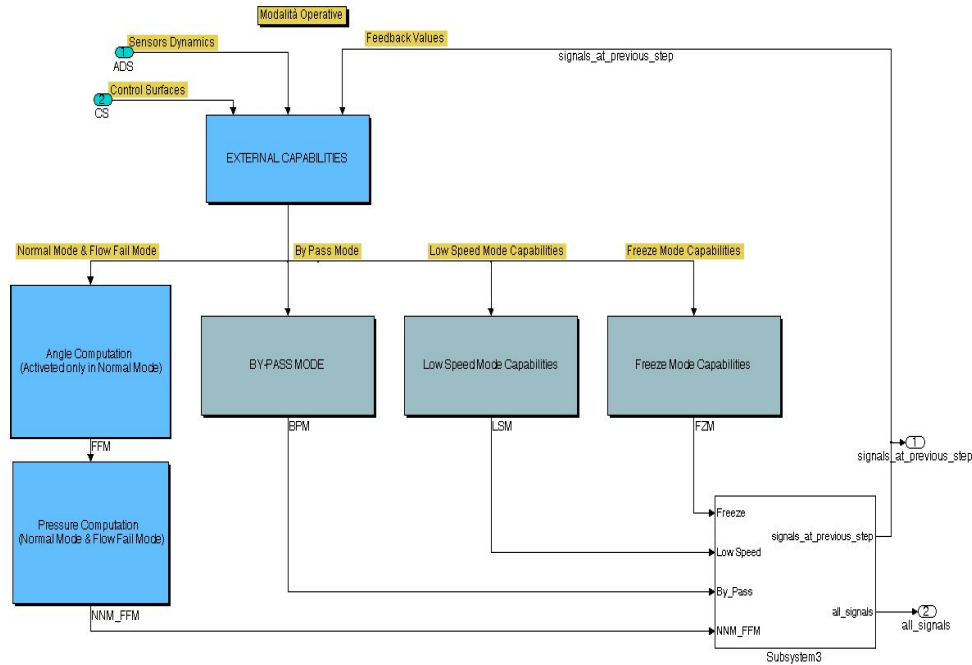


Figura 2.14: Modello Simulink delle modalità operative.

## 2.9 L'ambiente Matlab/Stateflow

Nei paragrafi che seguono viene presentata un'applicazione del software *Stateflow* di *Matlab*<sup>®</sup> alla realizzazione delle modalità operative della procedura di elaborazione dati aerea precedentemente introdotte.

Come mostrato in Fig. 2.8, sono state utilizzate due differenti programmazioni per realizzare la procedura di elaborazione, una tradizionale in *C++* ed una in ambiente *Stateflow*: i rispettivi segnali d'uscita vengono confrontati a valle tramite il blocco "Visualizzazione parametri ricostruiti".

L'aspetto innovativo, sottolineato nel presente lavoro, di una programmazione in ambiente *Stateflow* consiste nella possibilità di visualizzare graficamente gli stati attivi di un sistema anche molto complesso: ciò permette di ridurre i tempi di ottimizzazione della procedura e quelli di calcolo, rispetto ad una programmazione tradizionale in ambiente *Matlab*<sup>®</sup>, poiché gli stati non attivi non occupano risorse.

### 2.9.1 Dal concetto di automa alla teoria delle macchine a stati finiti

L'interesse che si è sviluppato nei confronti degli automi, ovvero dei dispositivi o degli oggetti che eseguono una particolare funzione in modo autonomo sulla base degli stimoli o degli ordini ricevuti, è dovuto alla possibilità di studiare facilmente la logica del loro comportamento, anche se complesso, rappresentandola attraverso grafi o matrici chiamate tabelle di verità. Le *truth tables* visualizzano le relazioni tra ingressi, uscite e stati dell'automata. Bisogna notare che lo stato di un automa è in realtà uno *stato di memoria* poiché è legato a tutto ciò che è successo in precedenza.

L'automata stabilisce il suo comportamento, quindi lo stato da attivare e l'uscita da fornire, sulla base dell'*input* che riceve e dello stato in cui si trova secondo lo schema:

$$(stato, input) \mapsto (nuovo\ stato, output)$$

Gli automi con un numero limitato di stati, ovvero che hanno memoria limitata, vengono definiti automi a stati finiti o macchine a stati finiti (*Finite State Machine*, FSM).

Più precisamente con questo termine è possibile indicare un qualunque sistema dinamico complesso il cui comportamento è definito da un insieme di stati e transizioni tra essi e da un numero di ingressi ed uscite finiti, che evolve nel tempo in base agli eventi che si verificano (*event-driven system* o anche *reactive system*).

In particolare la transizione del sistema da uno stato (modo) all'altro avviene se la condizione che rende possibile tale passaggio è vera.

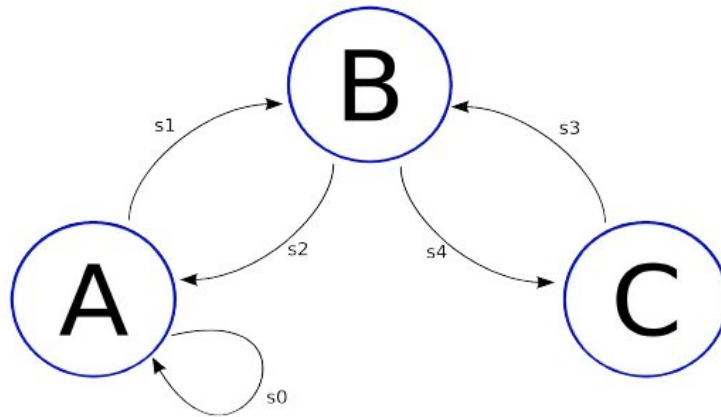
Per esempio un'automobile con cambio automatico può essere schematizzata con una macchina a stati finiti con un certo numero di situazioni possibili (parcheggio, movimento, retromarcia, ... ) e la transizione del sistema da una situazione all'altra avviene quando il guidatore muove lo *stick*.

Bisogna sottolineare che la rappresentazione di un sistema complesso tramite la teoria delle macchine a stati finiti non è sempre una proprietà intrinseca del sistema stesso ma è un modo semplificato e non riduttivo per poterlo studiare.

Generalmente le relazioni tra ingressi, uscite e stati di una FSM possono essere rappresentate da una tabella di verità.

Un altro approccio è rappresentato dai diagrammi di stato (*state diagrams* o *state transition diagrams*, STDs) e dai *bubble diagram* in Fig. 2.15.

Ogni stato o modo del sistema viene raffigurato con una bolla, da cui escono delle frecce che rappresentano tutte le transizioni possibili. Ogni freccia è quasi sempre associata ad etichetta (*label*) che indica la condizione o control-



**Figura 2.15:** Esempio di *bubble diagram*.

lo che fa transire lo stato: è chiaro che questo tipo di rappresentazione non ha limiti grafici ma ovviamente non deve lasciare ambiguità di interpretazione poiché ciò comporterebbe ambiguità di funzionamento della macchina a cui il diagramma corrisponde.

Nel 1983 David Harel<sup>[16]</sup> presentò una versione modificata degli *state transition diagrams* chiamati *statechart*.

Quest'ultimi forniscono infatti notevoli miglioramenti rispetto ai diagrammi delle transizioni: è possibile ad esempio la gestione della gerarchia tra stati, il parallelismo ovvero l'attivazione contemporanea di due o più stati, le giunzioni o collegamenti ed inoltre il comando *history* permette di tenere in memoria l'ultimo stato attivo e far ripartire il sistema da questo.

Il *software Stateflow* incorpora dunque i benefici sia dei diagrammi delle transizioni sia degli *statechart* e vi aggiunge delle innovazioni come la gestione dei diagrammi di flusso e la generazione efficiente di codice C.

Il *software* permette dunque di :

- progettare graficamente e simulare sistemi basati sulla FSM (*reactive systems*);
- sviluppare sistemi di controllo e supervisione;
- consente di modificare in modo facile e veloce il sistema in oggetto, valutarne il suo comportamento ed ottimizzarlo già in fase di progettazione.

Dunque è possibile utilizzare il *software* per creare intelligenze artificiali all'interno di moderni videogiochi, dove l'intelligenza artificiale indica il modo



in cui un oggetto virtuale interagisce e/o risponde agli stimoli indotti da altri oggetti o dallo stesso giocatore. Rende inoltre possibile scenari sempre più realistici, attivandoli in base ai comportamenti ovvero alle decisioni del giocatore: maggiore è il numero di eventi possibili a seguito di una decisione, maggiore è il realismo del prodotto.

Queste caratteristiche rendono inoltre *Stateflow* ideale per lo sviluppo dei sistemi di controllo di tipo *embedded* nel campo automobilistico (sistemi di condizionamento, ABS, ...), nell'automazione industriale e nel campo aerospaziale (sistemi di guida inerziale, sistemi per il controllo del volo di velivoli e missili, ...).

### 2.9.2 Stateflow: aspetti progetturali

Il flusso logico di progetto di un diagramma *Stateflow* è costituito tipicamente da alcuni passi fondamentali e coinvolge i seguenti componenti:

- creazione di un modello *Simulink*;
- creazione del diagramma *Stateflow*, tramite un *Editor* grafico;
- definizioni degli stati, delle transizioni ed inserimento dei dati tramite un *Explorer*;
- simulazione del sistema e generazione del codice C equivalente, attraverso il *Debugger* e il *Code Generation*.

Bisogna sottolineare che uno dei punti di forza di questo *software* è la possibilità di interagire con *Matlab* e *Simulink* sia in fase di modellazione che di simulazione ed analisi del sistema: i diagrammi *Stateflow* possono esser infatti incorporati in modelli *Simulink*.

Per la costruzione di un diagramma si utilizza un'interfaccia grafica (GUI) che permette di selezionare e trascinare gli oggetti nel foglio di lavoro.

L'aspetto tipico di un diagramma *Stateflow* è mostrato in Fig. 2.16.

Il formalismo utilizzato è quello di Harel in cui i rettangoli corrispondono agli stati (*chart*) del sistema, collegati tra loro mediante delle frecce che indicano le transizioni: la logica di funzionamento di questo *software* è detta infatti "*logica di salto*". Ogni transizione è etichettata da tre elementi:

$$\text{Evento}[\text{Condizione}]/\text{Azione}$$

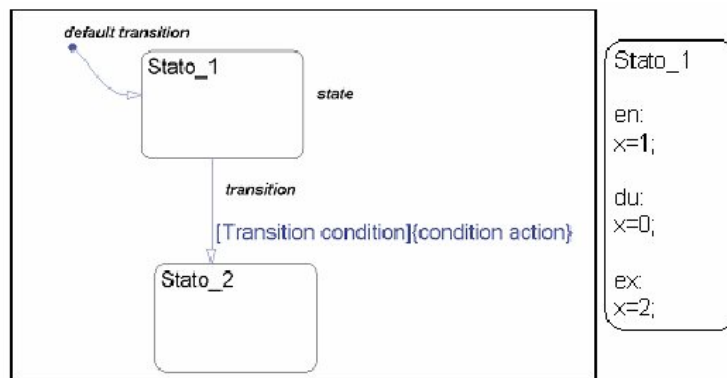
Ad esempio il passaggio dallo *Stato\_1* allo *Stato\_2* è possibile quando si verifica la condizione indicata in parentesi quadre (Condizione o *Transition condition*).

Nel momento in cui si verifica tale passaggio il sistema esegue l'operazione in parentesi graffe (Azione o *condition action*). Ogni transizione che riporta al

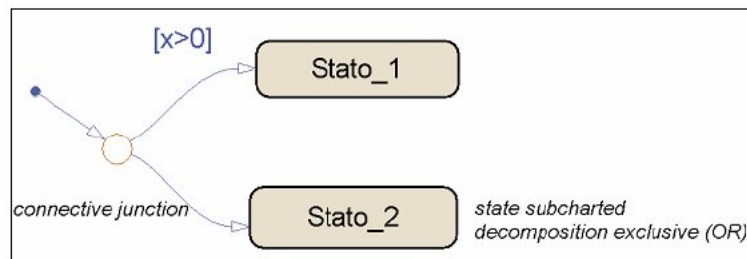
medesimo stato del sistema è detta *self-transition*.

Il punto d'ingresso in un diagramma *Stateflow* è definito *default transition*. E' possibile far eseguire all'interno di uno stato alcune operazioni (a destra in figura) ogni volta che lo stato viene attivato (comando *entry*), durante il periodo in cui lo stato rimane attivo (comando *during*) e all'uscita da esso (comando *exit*).

La *connective junction* o punto decisionale è assimilabile ad un ciclo *if* di



**Figura 2.16:** Stati e transizioni in un diagramma *Stateflow*.



**Figura 2.17:** Diagramma *Stateflow*: la *connective junction*.

*Matlab*. Ogni linea uscente dal punto decisionale rappresenta un possibile percorso del sistema: il numero di condizioni logiche che è possibile associare deve essere però il minimo possibile (Fig. 2.17).

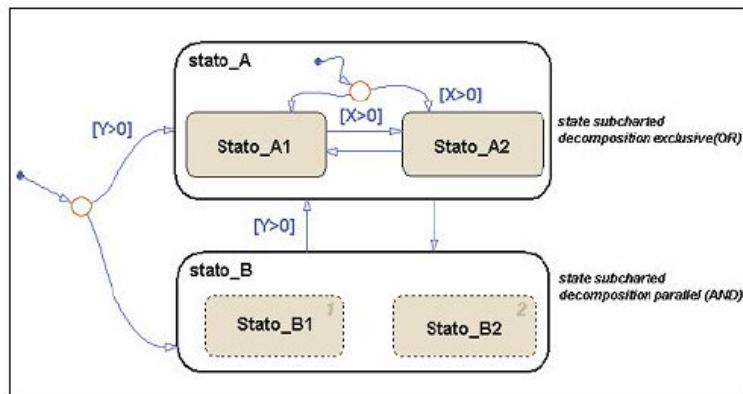
E' possibile attivare gli stati in due modalità.

Con riferimento alla Fig. 2.18, se si verifica la condizione  $[Y > 0]$  si attiverà lo *stato\_A* da cui accederemo allo *stato\_A1* se  $[X \leq 0]$  oppure allo *stato\_A2* se  $[X > 0]$ . In questo caso si parla di attivazione in modo esclusivo tramite

la *decomposition or*, rappresentata in linea continua.

Se la condizione  $[Y > 0]$  non si verifica, si accede allo *stato\_B* da cui verranno attivati, senza che si verifichi nessuna ulteriore condizione, *stato\_B1* e *stato\_B2* in parallelo tramite la *decomposition and*: in questo caso le *chart* sono delimitate graficamente da una linea tratteggiata ed i numeri in alto a destra indicano l'ordine temporale di esecuzione.

Nel caso di sistemi caratterizzati da logiche di funzionamento complesse,



**Figura 2.18:** Diagramma *Stateflow*: attivazione *and/or*.

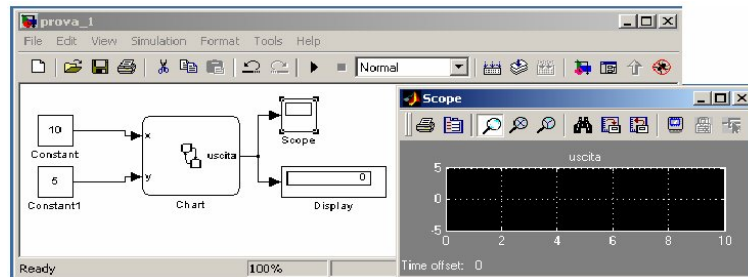
con un numero notevole di stati e transizioni tra essi, è possibile rendere il modello più leggibile, nascondendo il contenuto della *chart* con il comando *subcharted*.

Tutte le variabili utilizzate all'interno di un diagramma devono essere definite ed inserite tramite l'*Explorer* nella barra degli strumenti (i dati possono essere presi da *Simulink* o possono essere di tipo locale ovvero interni al diagramma). È importante definire correttamente le variabili poiché vengono trattate diversamente a seconda della tipologia: ad esempio ad un evento che provoca una transizione è associata un'informazione qualitativa mentre ad un dato una di tipo quantitativo.

I risultati di una generica simulazione possono essere visualizzati con degli *scope* o dei *display* di *Simulink* (Fig. 2.19).

### 2.9.3 Applicazione del toolbox alle logiche di gestione delle ridondanze

In questo paragrafo viene descritta la realizzazione delle logiche di *monitoring* e *voting* degli angoli e delle pressioni, tramite il *toolbox Stateflow* di *Matlab*. La Fig. 2.20 schematizza il *level 0* del processo di elaborazione degli angoli: si può individuare un blocco contenente le logiche di computazione ed in cascata



**Figura 2.19:** Diagramma *Stateflow*: visualizzazione dei risultati.

i blocchi in cui sono implementate rispettivamente le logiche di *monitoring* e *voting*.

Per quanto riguarda il *monitor* bisogna distinguere due blocchi:

- *Normal Monitoring AoA/AoS*: relativo alla condizione di nessuna avaria individuata allo *step* di esecuzione precedente dell'algoritmo e che dunque ha l'obiettivo di individuare una possibile prima avaria;
- *One Fail Monitoring AoA/AoS*: attivato quando si è già verificata un'avaria, con il compito di individuarne un'eventuale successiva;

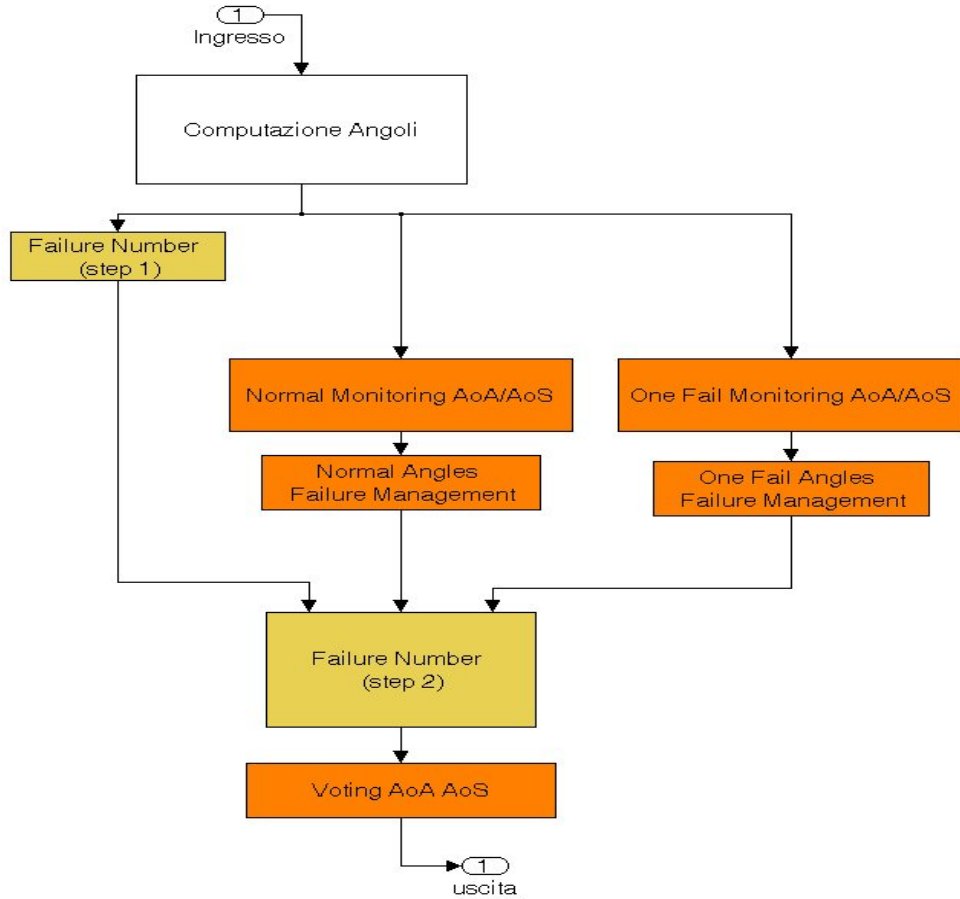
I blocchi "*Normal Angles Failure Management*" e "*One Fail Angles Management*", posti in cascata, aggiornano opportune variabili *booleane* (non esaminate però in questo lavoro) in base alla persistenza di un'avaria.

E' chiaro che sia il processo di *monitor* che quello di *voting* sono legati al numero di sonde in avaria: ciò rende necessaria la presenza del blocco "*Failure number (step1)*" che ha il compito di fornire in uscita un parametro (*numero\_avarie*) indicativo del numero di avarie rilevate dal sistema.

Inoltre, prima del *voting*, occorre inserire il blocco "*Failure number (step2)*" che si comporta come un vero e proprio *switch* ovvero, sulla base del parametro *numero\_avarie*, gestisce i segnali provenienti dal *monitoring* e li fornisce in ingresso al *voting*.

Dopo aver esaminato brevemente le funzioni svolte dai diversi blocchi che compaiono nel *level 0* del processo di elaborazione angoli, si passa ad una descrizione dettagliata dell'implementazione in *Stateflow* delle logiche di *monitoring*.

La Fig. 2.21 mostra la realizzazione del "*Normal Monitoring AoA/AoS*": si osservi la presenza di quattro *chart* eseguite in parallelo (*decomposition and*) secondo l'ordine indicato dai numeri posti in alto a destra e relative ai quat-



**Figura 2.20:** Procedura di elaborazione angoli.

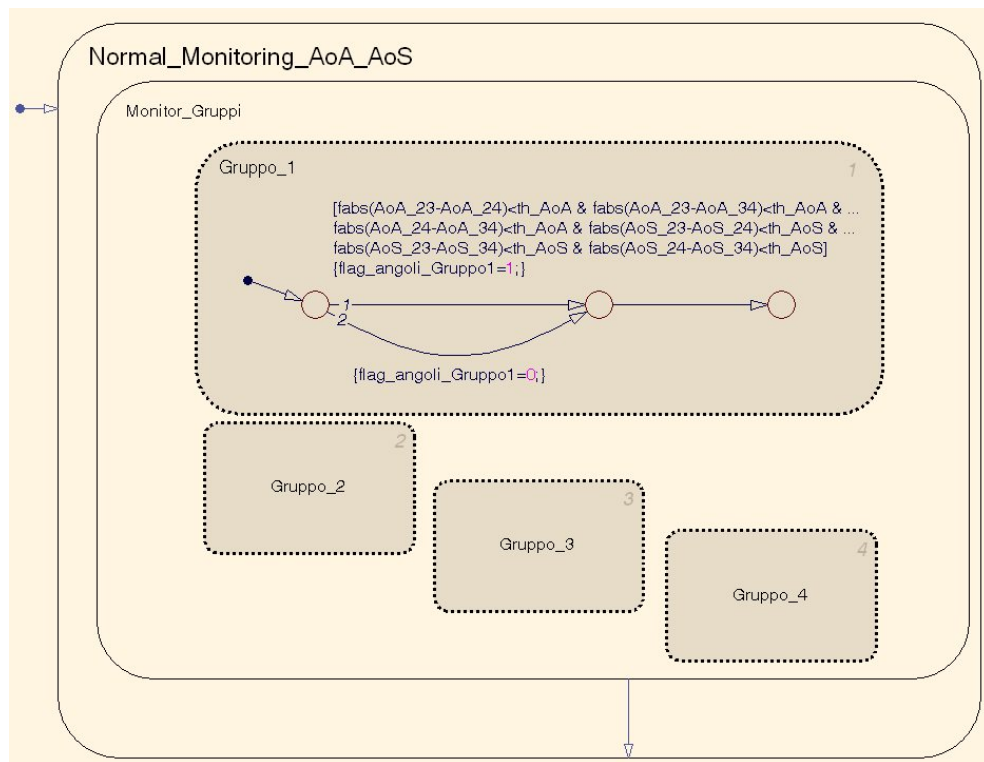
tro gruppi descritti in 2.8.2.

Facendo riferimento al *Gruppo\_1* che contiene le coppie  $(\alpha_{23}, \alpha_{24}, \alpha_{34})$  se tutte le differenze in valore assoluto (comando *fabs*) tra le coppie risultano inferiori alla soglia *th\_AoA*, il programma esegue l'azione contenuta in parentesi graffe ovvero assegna alla variabile booleana *flag\_angoli\_Gruppo1* il valore 1 (1 accettabile, valido), in caso contrario se la condizione tra parentesi quadre non è rispettata verrà attivato il percorso in basso (indicato dal numero 2) e la variabile assumerà valore 0 (non accettabile, non valido).

Si osservi che il numero di condizioni di transizione è quello strettamente necessario. La sintassi è dunque simile a quella di una programmazione in C++ ma rispetto a quest'ultima *Stateflow* consente di realizzare i cicli *if* graficamente, facendo partire da un punto decisionale due percorsi, e ne verifica

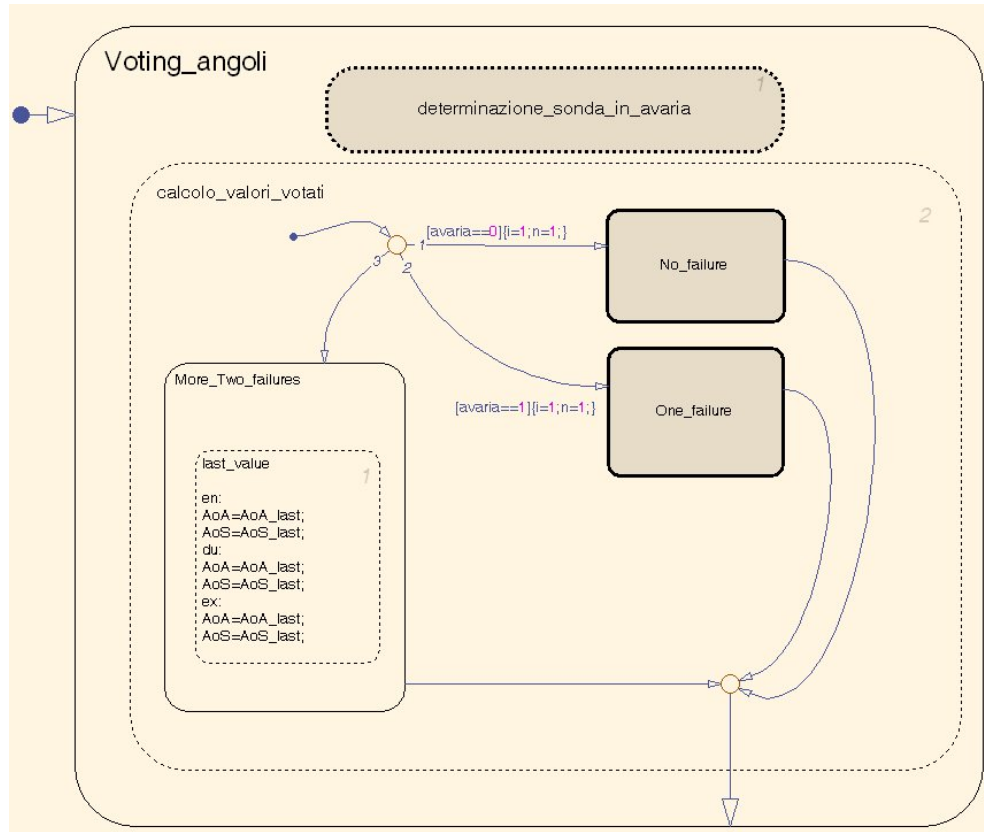
il corretto funzionamento in modo semplice ed immediato tramite l'uso dei colori.

In modo analogo si procede per l'angolo di *sideslip*  $\beta$  e per i gruppi 2, 3, 4. L'algoritmo di *voting* prevede dunque due *chart* principali. Nel caso di sistema pienamente funzionante verrà attivata quella denominata "*No\_failure*" ed i valori degli angoli verranno calcolati come media tra i valori centrali, dopo aver ordinato le sei stime disponibili. In presenza di una sonda in avaria tali stime si riducono a tre e dunque il valori votato coincide col centrale. Se il numero di avarie è maggiore il sistema, mantiene l'ultimo valore calcolato poiché non è in grado di fornire stime consolidate. Per poter conoscere il numero di avarie presenti (ovvero il valore del parametro interno "*avaria*") occorre posizionare a monte una *chart* appositamente dedicata, denominata "*determinazione\_sonde\_in\_avaria*".



**Figura 2.21:** Procedura di elaborazione angoli.

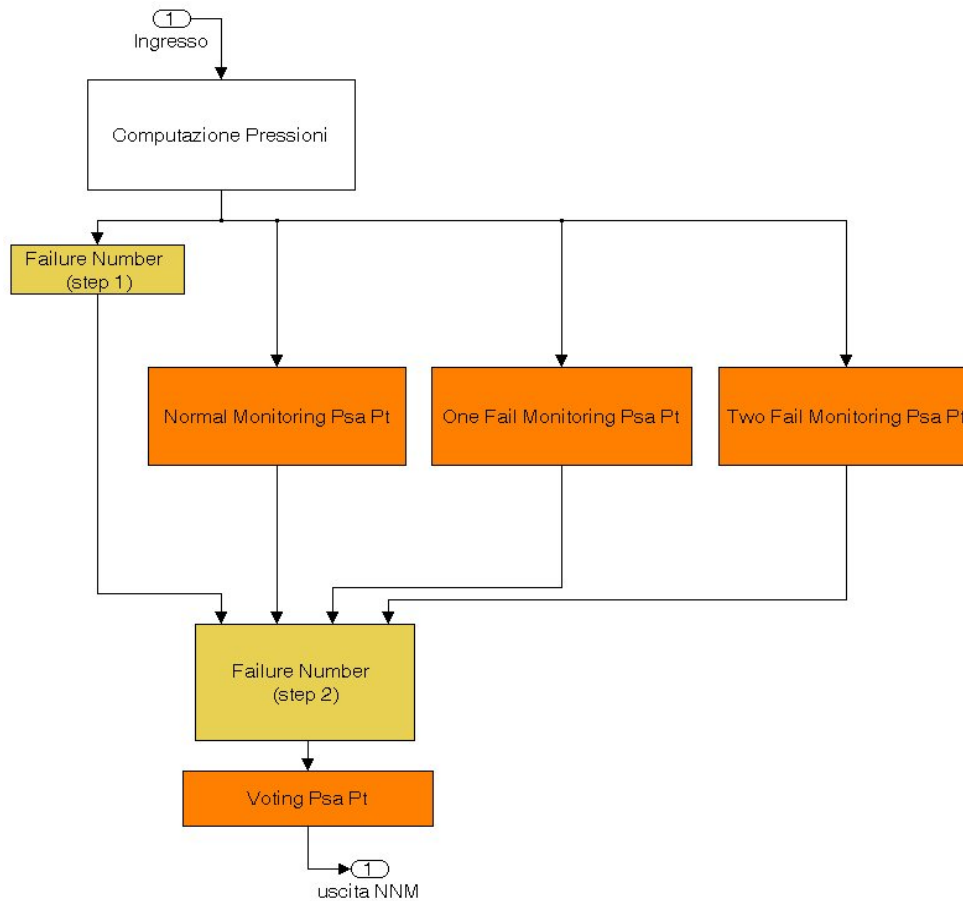
La procedura di elaborazione delle pressioni può esser schematizzata come in Fig. 2.23.



**Figura 2.22:** Procedura di elaborazione angoli.

Confrontando quest'ultima con la Fig. 2.20 è possibile notare una fase di *monitor* su  $P_{sa}$  e  $P_t$  anche in presenza di due sonde in avaria. Come detto in precedenza infatti, il sistema è capace di garantire una quadruplice ridondanza (*two fail operative - fail safe*) per quanto riguarda il processo di ricostruzione delle pressioni.

Di conseguenza, l'algoritmo di *voting* presenterà una *chart* aggiuntiva, denominata "*Two\_failures*", in basso a destra in Fig. 2.24. In questo caso i parametri votati sono calcolati semplicemente come media tra i valori forniti dalle uniche due sonde funzionanti. La figura evidenzia inoltre una potenzialità del *software Stateflow*, che consiste nel visualizzare graficamente lo stato attivo e dunque il percorso compiuto, tramite una linea blu in grassetto: ciò consente, istante per istante, di conoscere, controllare ed ottimizzare il comportamento del sistema.



**Figura 2.23:** Procedura di elaborazione pressioni.

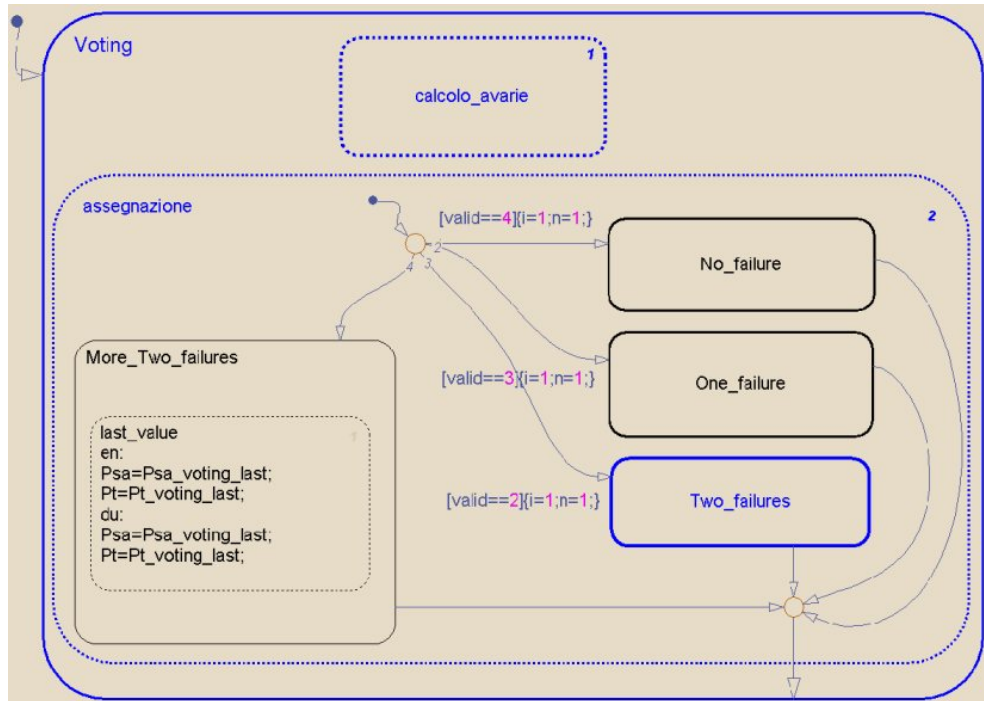
#### 2.9.4 Vantaggi della programmazione in Stateflow

L'attività svolta consente di affermare che il *software Stateflow* fornisce un ambiente grafico per la modellazione avanzata di sistemi e processi che evolvono nel tempo in base agli eventi che si verificano (FSM), ambiente nel quale è possibile creare in modo interattivo il diagramma degli stati e delle transizioni tra quest'ultimi.

La logica di funzionamento alla base di questo *software* viene infatti definita "*logica di salto*".

E' da sottolineare la possibilità di individuare e monitorare lo stato interno del sistema attualmente in uso: ciò consente di simulare il comportamento di un ipotetico utente, visualizzando quindi tutte le azioni che intraprende.





**Figura 2.24:** Attivazione del voting sulle pressioni in presenza di due avarie.

Infatti mentre una simulazione è attiva, *Stateflow* riflette graficamente le transizioni tra stati mediante l'uso dei colori (i rami o gli stati attivi sono colorati in blu).

Rispetto al formalismo delle reti di Petri consente non solo la simulazione dell'andamento del sistema ma anche la registrazione e l'analisi del transitorio tra uno stato e l'altro.

Uno degli elementi qualificanti di questo *software* è dato inoltre dalla possibilità di rappresentare in modo innovativo su uno stesso foglio, con un formalismo grafico snello, elegante e chiaro, i diversi modi operativi in cui può trovarsi un sistema anche complesso come quello per la ricostruzione dei dati aria di un velivolo *Fly-by-Wire*.

Inoltre facilita l'ottimizzazione del sistema rendendone possibile l'integrazione già in fase di progettazione. Occorre evidenziare che, rispetto ad una programmazione tradizionale, gli stati ed i percorsi non attivi non occupano risorse e ciò comporta chiaramente una riduzione anche dei tempi di calcolo.

## 2.10 Analisi dell'errore tramite simulazioni alla Monte Carlo

Una volta completato il modello è possibile lanciare la simulazione: ciò genera per mezzo dello *Stateflow Coder* un codice C *stand-alone* ovvero una *S-function* di *Simulink* che è possibile usare in *real time* in un gran numero di ambienti per la prototipazione rapida o per le simulazioni *hardware-in-the-loop*.

I risultati vengono visualizzati tramite *scope* o *display* di *Simulink*. L'utilizzo di due programmazioni indipendenti ha permesso di ridurre i tempi di ottimizzazione e revisione delle programmazioni stesse. A tal proposito è stata condotta un'ampia campagna di verifica della correttezza del procedura realizzata, in diverse condizioni di volo e tenendo conto dell'eventuale presenza di sonde in avaria.

Successivamente si è proceduto all'analisi dell'errore commesso sui parametri votati, tramite un approccio alla Monte Carlo.

Questo termine fu coniato all'inizio della seconda guerra mondiale da J. von Neumann e da S. Ulam mentre lavoravano al progetto Manhattan presso il centro di ricerche nucleari di Los Alamos. I due ricercatori utilizzarono l'espediente di sostituire le equazioni che descrivono la dinamica delle esplosioni nucleari con un insieme di numeri casuali ed indicarono il lavoro che stavano svolgendo con un nome in codice, prendendo ispirazione dal celebre principato monegasco e dalla aleatorietà dei risultati che si possono riscontrare presso la sua casa da gioco.

Questo metodo fa parte della famiglia dei metodi statistici non parametrici: è utile per risolvere problemi computazionali legati a "*test esatti*", ad esempio metodi basati su distribuzioni binomiali. Il metodo è usato per trarre stime attraverso delle simulazioni: si basa su algoritmi che generano serie di numeri tra loro indipendenti, che seguono la distribuzione di probabilità del fenomeno da indagare. La non correlazione tra i numeri è assicurata dal test del  $\chi^2$ .

Nel presente lavoro è stata condotta dunque un'analisi statistica dell'errore, che consiste nel perturbare i segnali in ingresso alla procedura di elaborazione (ovvero  $P_{fronti}$ ,  $P_{slot2}$ ,  $\lambda_i$ ,  $\lambda_j$ ) sovrapponendo ad essi opportune incertezze, caratterizzate da una distribuzione di tipo gaussiano di parametri  $\mu = 0$  e  $3\sigma$ . Secondo tale criterio, nel 99,7% dei casi il segnale generato appartiene all'intervallo  $[\mu - 3\sigma, \mu + 3\sigma]$ .

Analizzando i risultati ottenuti a seguito di tale campagna di prove, si è giunti ad una definizione preliminare delle soglie di monitor  $th_{Psa}$  e  $th_{Pt}$ , utilizzate all'interno della stessa procedura di elaborazione: tali soglie verranno comunque calibrate in fase in prove in volo (*Flight Test*).

Prima di individuare gli errori sui parametri ricostruiti occorre però che ogni punto dell'involuppo di volo, su cui si va ad effettuare tale calcolo, venga tenuto costante per un certo numero di istanti (*frame*) in modo da evitare che le grandezze statistiche dipendano dal frame stesso. E' stata effettuata dunque una serie di simulazioni preliminari in diverse condizioni di volo ( $\alpha$ ,  $\beta$ , M, quota) per scegliere quell'opportuno valore di *frame*, oltre il quale gli errori sui parametri votati risultano costanti.

Questi sono cautelativamente calcolati in termini di  $\mu + 3\sigma$  ovvero, con riferimento all'angolo  $\alpha$ :

$$\begin{cases} \mu_{-\alpha} = \mu(\alpha_{-vot} - \alpha_{-sim}) \\ \sigma_{-\alpha} = \sqrt{(\alpha_{-vot} - \alpha_{-sim})^2} \end{cases} \quad (2.9)$$

La media e la deviazione standard vengono calcolate sulla differenza tra il valore votato cioè ricostruito e quello di simulazione del parametro di volo preso in esame. Analogamente per l'angolo di *sideslip*.

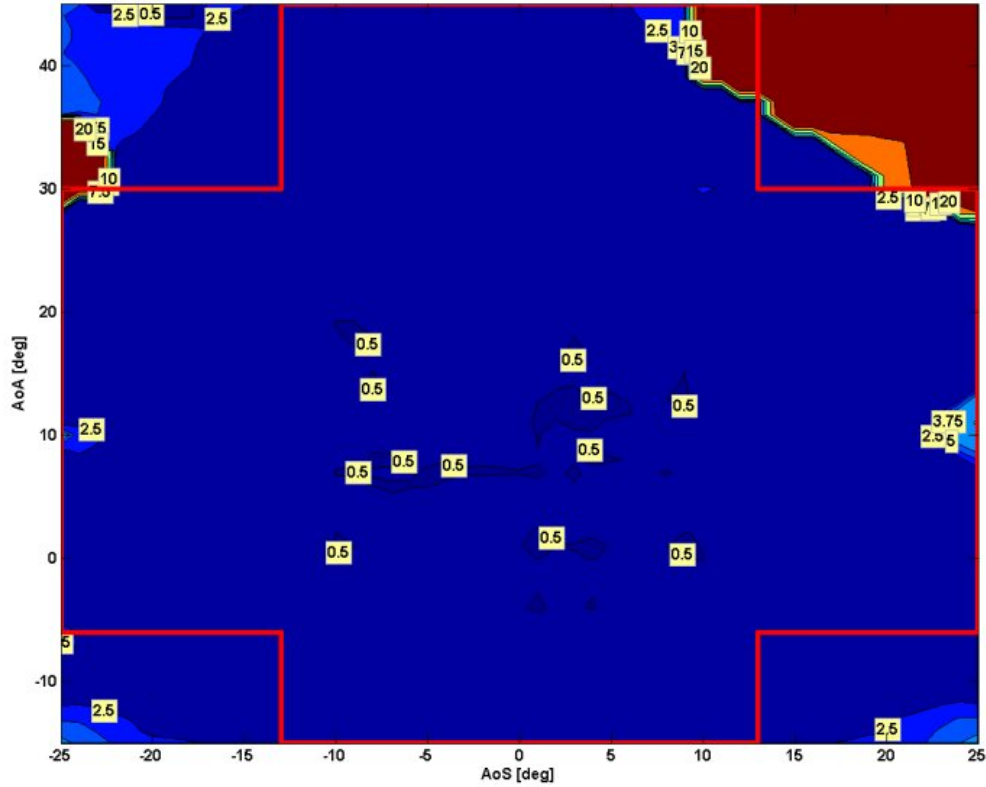
I grafici riportati nel presente lavoro sono ottenuti tramite il comando *contourf(z)* di *Matlab* che permette di graficare, sottoforma di curve di livello o isolinee, gli errori sui parametri di volo. I dati raccolti durante la generica simulazione vengono inseriti come matrici ( $z$ ) e le aree tra curve di uguale valore vengono riempite con lo stesso colore: il numero che identifica ogni curva può esser inserito tramite il comando *clabel*.

Nel presente lavoro di tesi, si è scelto di riportare le mappe d'errore commesso nella ricostruzione dell'angolo di incidenza, indicato con *AoA*, e di *sideslip*, indicato con *AoS*, in tutto l'involuppo di volo relativo al Mach considerato ( $-15^\circ < \alpha < 45^\circ$ ,  $-25^\circ < \beta < 25^\circ$  a Mach = 0.4).

In realtà è stata condotta una campagna piuttosto dettagliata di valutazione di tali errori per diversi valori del numero di Mach, per diverse condizioni di volo quali crociera, manovra, decollo, carrelli estratti, superfici di volo deflesse: inoltre le analisi sono state eseguite in condizione di piena funzionalità del sistema sonde ma anche in presenza di una o due sonde in avaria, prendendo in esame tutte le possibili combinazioni tra esse.

Bisogna sottolineare che l'involuppo di volo è funzione del numero di Mach ed in generale tende a restringersi all'aumentare di quest'ultimo: la linea rossa riportata nei grafici è in realtà solo una ulteriore limitazione che indica il reale involuppo ammissibile (*permitted range*).

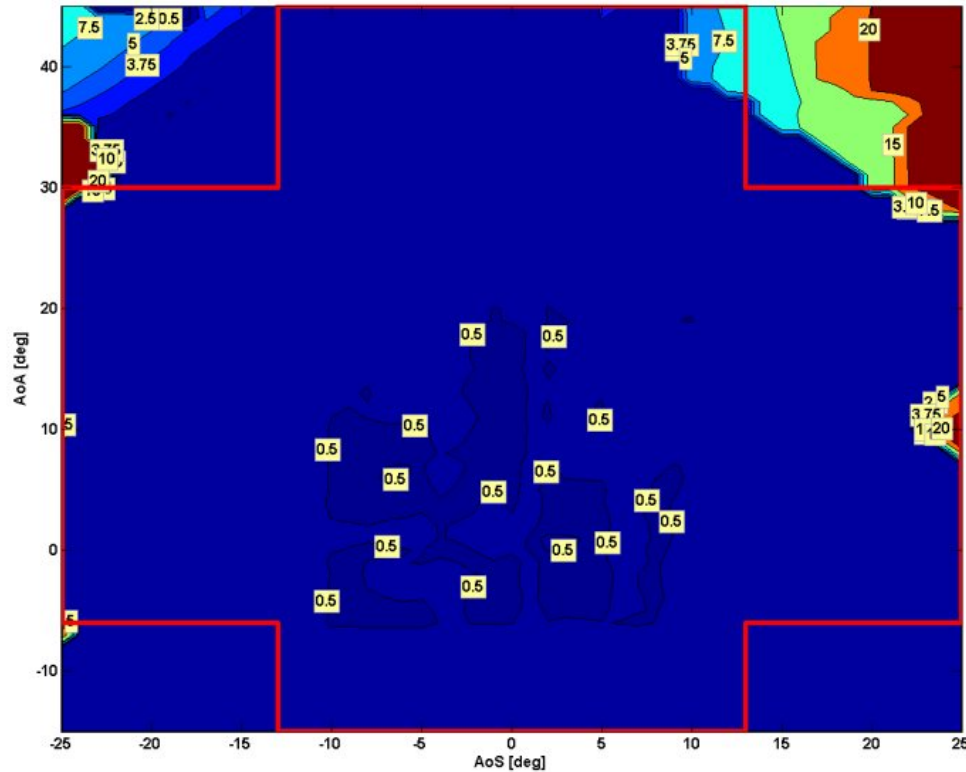
Si è ritenuto significativo riportare l'andamento dell'errore su *AoA* ed *AoS* a Mach 0.4, relativamente alla condizione di volo di crociera, sia in condizione di piena funzionalità del sistema dati aria (vedi Fig. 2.25 e Fig. 2.26), sia in presenza di una sonda in avaria (vedi Fig. 2.27 e Fig. 2.28): in questo



**Figura 2.25:** Mappa dell'errore in  $^{\circ}$  commesso su  $\alpha$ , a Mach 0.4, in condizione di piena operatività.

caso però, per ogni punto dell'involuppo di volo è indicato l'errore massimo (denominato *worst case*) che si ottiene mandando in avaria rispettivamente la sonda 1, 2, 3, 4.

I grafici forniscono dunque un quadro cautelativo del degrado delle prestazioni del sistema dati aria in presenza di sensori in avaria, in diverse condizioni di volo e numero di Mach. Inoltre risultano particolarmente significativi poiché rappresentano una verifica della bontà della procedura di elaborazione sviluppata in ambiente *Stateflow*: un errore elevato, in particolar modo in condizioni di piena funzionalità delle sonde, è indice di uno scostamento elevato del parametro ricostruito rispetto al suo valore di simulazione. In conclusione i risultati ottenuti possono esser ritenuti dunque soddisfacenti.

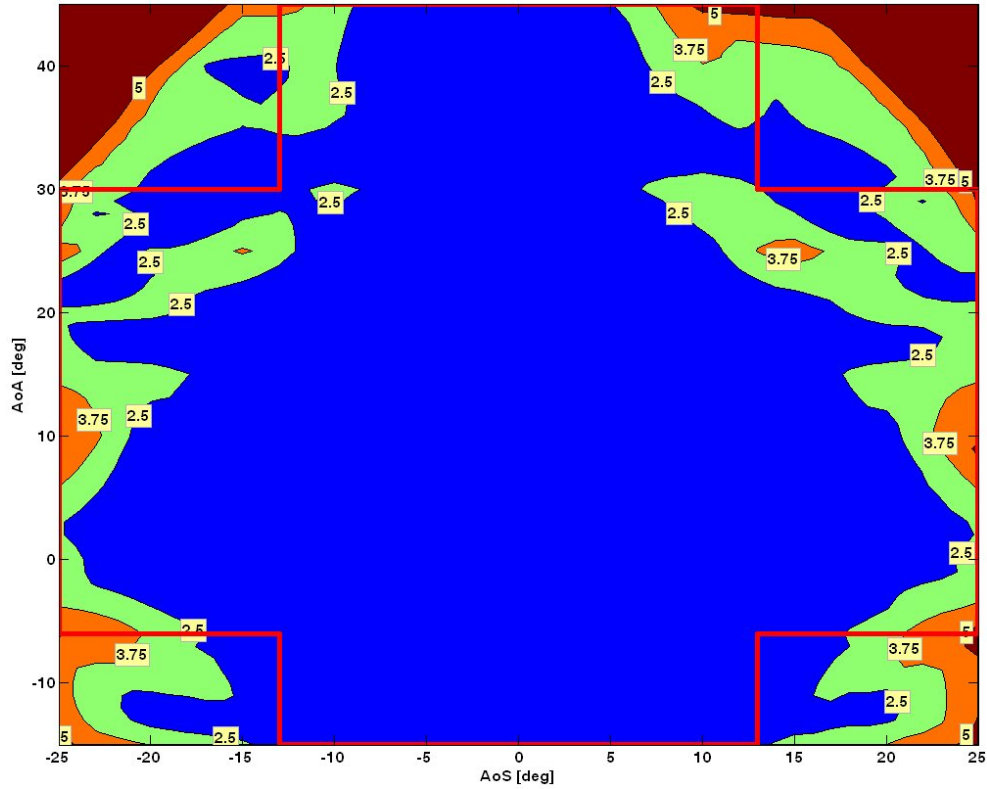


**Figura 2.26:** Mappa dell'errore in  $^{\circ}$  commesso su  $\beta$ , a Mach 0.4, in condizione di piena operatività.

## 2.11 Aspetti innovativi dell'argomento di ricerca trattato

La determinazione dei dati aria e gli strumenti utilizzati a tale scopo, svolgono un ruolo importante nei moderni velivoli FBW, poiché vengono impiegati dai calcolatori di bordo per svolgere funzioni importanti, quali la protezione dell'involuppo e l'adeguamento dei guadagni dei sistemi di controllo al mutare delle condizioni di volo.

Inoltre nei velivoli militari o velivoli non pilotati vengono impiegati nei sistemi che svolgono funzioni di *stall warning*, *weapons delivery*, *terrain avoidance* e *flight safety* mentre nell'aviazione civile, la separazione verticale del traffico aereo è possibile solo a seguito di un'accurata conoscenza di tali parametri. A riguardo, dal 1999 il DIA è impegnato in un programma di ricerca che propone lo sviluppo di un sistema di controllo di tipo FBW dei comandi primari di volo, con attuazione idraulica, del tipo di quello implementato sull'adde-



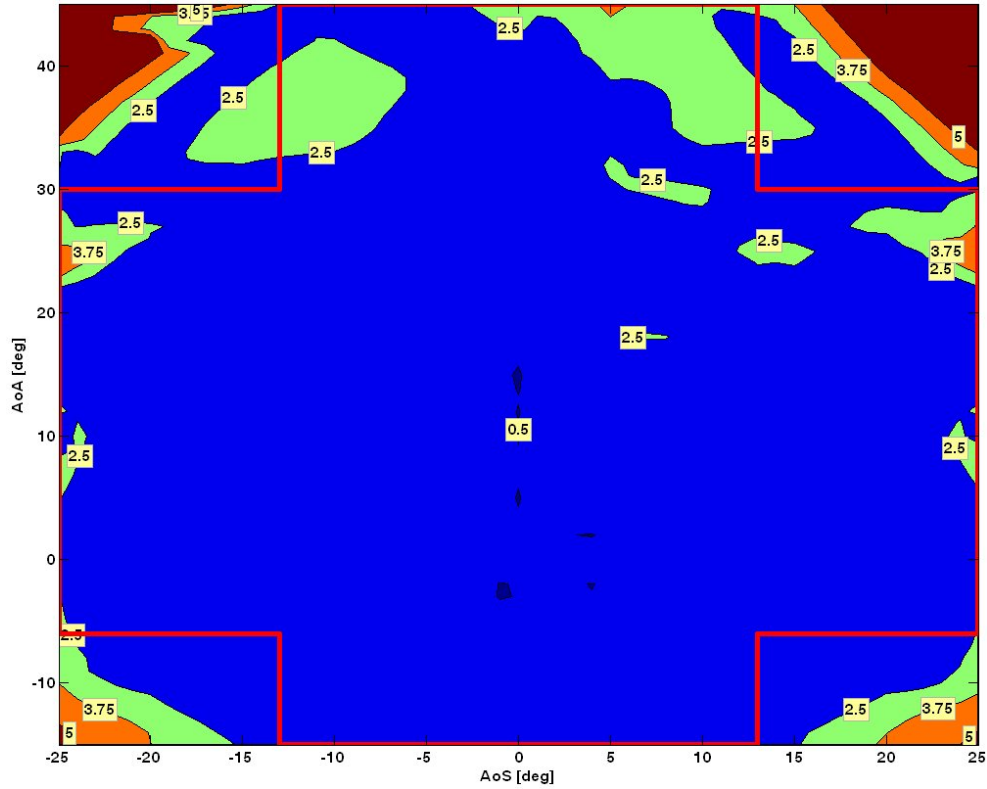
**Figura 2.27:** Mappa dell'errore in  $^{\circ}$  commesso su  $\alpha$ , a Mach 0.4, in presenza di una sonda in avaria.

stratore militare di nuova generazione Aermacchi M346.

Il presente dottorato ha contribuito allo sviluppo di metodologie innovative per l'elaborazione dei dati aria partendo dalle misure locali fornite da quattro sensori di tipo multi-funzione.

I sensori possono esser soggetti a diverse tipologie di avarie (bloccaggio dovuto ad occlusione parziale o completa delle prese di pressione, rottura delle palette di allineamento, ...) che comportano la perdita del velivolo o comunque l'interruzione della missione ed il successivo passaggio ad un modalità operativa degradata per il rientro del velivolo alla base. Tali considerazioni sottolineano la necessità di sviluppare logiche di individuazione delle avarie (algoritmi di *monitor*), logiche di gestione delle ridondanze (algoritmi di *voting*) e riconfigurazione del sistema.

In letteratura, al momento, vengono descritte tecniche di gestione delle ridondanze, ad esempio, relativamente alla determinazione della pressione a



**Figura 2.28:** Mappa dell'errore in  $^{\circ}$  commesso su  $\beta$ , a Mach 0.4, in presenza di una sonda in avaria.

partire dai segnali forniti da quattro sonde ma il problema oggetto della presente tesi è in realtà molto più complesso. Infatti, per quanto concerne la determinazione degli angoli di incidenza e derapata, è indispensabile disporre contemporaneamente dei segnali forniti da due sensori differenti. Le stime disponibili sono dunque sei: si tratta quindi di una gestione non-standard delle ridondanze ed in merito la letteratura fornisce scarsissime informazioni. Inoltre poiché è possibile il verificarsi di avarie sia dei sensori angolari che di pressione, occorre prevedere diverse modalità operative in funzione delle possibili combinazioni di guasto.

A seguito di tali considerazioni, il contributo dato dalla presente tesi di dottorato può ritenersi una risposta originale alle problematiche sopra esposte.

## Capitolo 3

### Sviluppo di una sonda multi pressione basata su sistemi di tipo FADS

Nell'ambito del presente lavoro di Dottorato è stato affrontato lo studio di tecniche alternative per la determinazione dei dati aria mediante l'impiego di sensori non intrusivi. Quest'attività trae origine dalle esperienze maturate presso il Dipartimento di Ingegneria Aerospaziale di Pisa (DIA), finalizzate alla definizione di metodi e modelli per l'integrazione dei dati aria nel *Flight Control System* (FCS) di velivoli *Fly-by-Wire* (FBW). La sonda progettata si ispira ai sistemi di tipo *Flush Air Data System* (FADS), ovvero sistemi di sensori dati aria di tipo non intrusivo, composti da una serie di prese di pressione realizzate direttamente sulla parte prodiera della fusoliera, le cui misure vengono elaborate da algoritmi di calcolo dedicati.

Il numero elevato di prese di pressione ne consente il funzionamento anche in caso di perdita di una o più misure dovute al verificarsi di avarie. Quest'aspetto rende i FADS particolarmente adatti a sistemi di tipo FCS-FBW, caratterizzati da architetture a molteplice ridondanza.

Inoltre la possibilità di posizionare "a pelle", direttamente sulla fusoliera, le prese di misura, consente l'installazione dei FADS su velivoli da rientro spaziale come l'X-33, che devono esser privi di appendici soggette ad eccessivo riscaldamento in fase di rientro in atmosfera, velivoli non pilotati, che richiedono impianti a basso costo ed ingombri limitati, velivoli *stealth* poiché tali sistemi soddisfano l'esigenza di *low observability*.

La stima dei parametri di volo a partire da un numero elevato di segnali disponibili rappresenta un problema piuttosto complesso e comporta la necessità di monitorare tali segnali per individuare eventuali avarie.

Negli ultimi tempi si è sviluppato un notevole interesse, anche dal punto di vista industriale, nei confronti di tale tipologia di sensori: ne è testimonianza



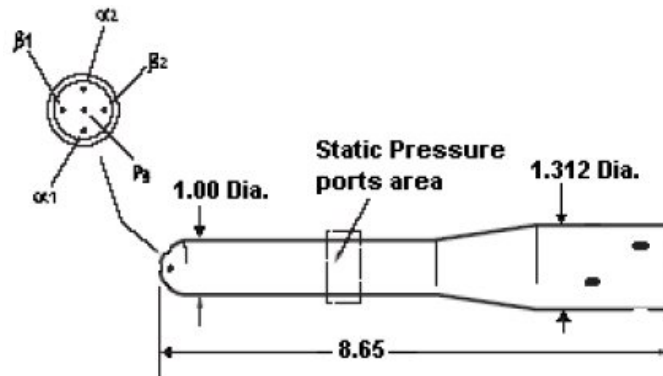
ad esempio il progetto europeo nEUROn, per lo sviluppo di un velivolo da combattimento non pilotato dalle accentuate caratteristiche *stealth*. Il contributo tecnologico del nostro paese verterà tra le altre cose proprio sullo sviluppo di un sistema FADS. Nonostante questo crescente interesse, la letteratura risulta ancora povera di lavori sull'argomento. Il presente dottorato getta dunque le basi per una ricerca in tal senso. Il lavoro è suddiviso in due parti. Nella prima è stato condotto un vero e proprio studio di fattibilità: scelte alcune possibili geometrie, sulla base di forme già presenti in letteratura, queste sono state sottoposte a studi fluidodinamici (*Computational Fluid Dynamics*, CFD). Tali studi hanno portato alla definizione di una geometria preliminare e ad un relativo *database* di pressione locale.

Nella seconda parte del lavoro è stato affrontato il problema della ricostruzione dei parametri di volo tramite tecniche di intelligenza artificiale: tale approccio si presenta come una valida alternativa alle tecniche di tipo classico che utilizzano funzioni polinomiali di taratura. Il *database* preliminare, ottenuto nella prima parte, è stato utilizzato dunque per allenare le reti neurali. Successivamente è stato condotto uno studio di ottimizzazione del numero di prese di pressione, della struttura della rete e della tipologia di allenamento che ha permesso di raggiungere risultati soddisfacenti in termini di errore commesso sui parametri di volo.

L'architettura neurale realizzata è capace di soddisfare non solo alla necessità di ricostruire i parametri d'interesse ma anche di tener conto delle problematiche connesse all'impegno di memoria e alla rapidità di allenamento delle reti stesse.

### 3.1 Definizione della geometria

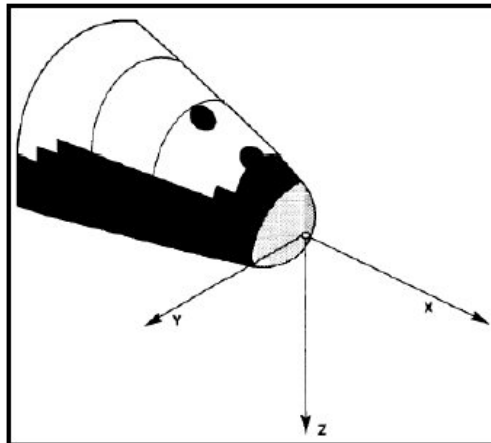
In letteratura sono presenti diversi tipi di sensori che utilizzano principi di funzionamento spesso molto differenti tra loro: per scegliere una possibile geometria della sonda sono state prese in esame diverse tipologie, integrandone le caratteristiche principali, con l'obiettivo di creare una geometria facilmente realizzabile dal punto di vista tecnologico. Le dimensioni della sonda fanno riferimento a quelle di un tubo di *Pitot*, una delle apparecchiature più semplici tra quelle presenti in commercio, ed inoltre analizzando le sonde multi funzione, riportate in Fig. 3.1, è stata valutata la possibilità di utilizzare un'estremità sferica per il posizionamento delle prese di misura. Il lavoro si ispira soprattutto ai sistemi di tipo FADS caratterizzati da un numero elevato (da 5 a 30) di prese di pressione realizzate direttamente sul *nose* del velivolo o sul *leading edge* dell'ala (in questo caso si parla di LE-FADS). Tali sistemi sono capaci di garantire dunque un opportuno livello di ridondanza fornendo misure attendibili anche in presenza di eventuali avarie o condizioni d'ombra.



**Figura 3.1:** Esempio di sonda multi funzione per il calcolo degli angoli di assetto (dimensioni in cm).

Trovano impiego tipicamente su velivoli da rientro spaziale e su velivoli sperimentali ad elevate prestazioni come l'F18 HARV (*High Research Vehicle*): in queste applicazioni tutte le prese di pressione sono realizzate sul *nose* e disposte come nelle figure Fig. 3.2 e Fig. 3.3. Le varie misure di pressione locale vengono poi inviate ad un'unità di calcolo dedicata che ricostruisce i parametri di volo.

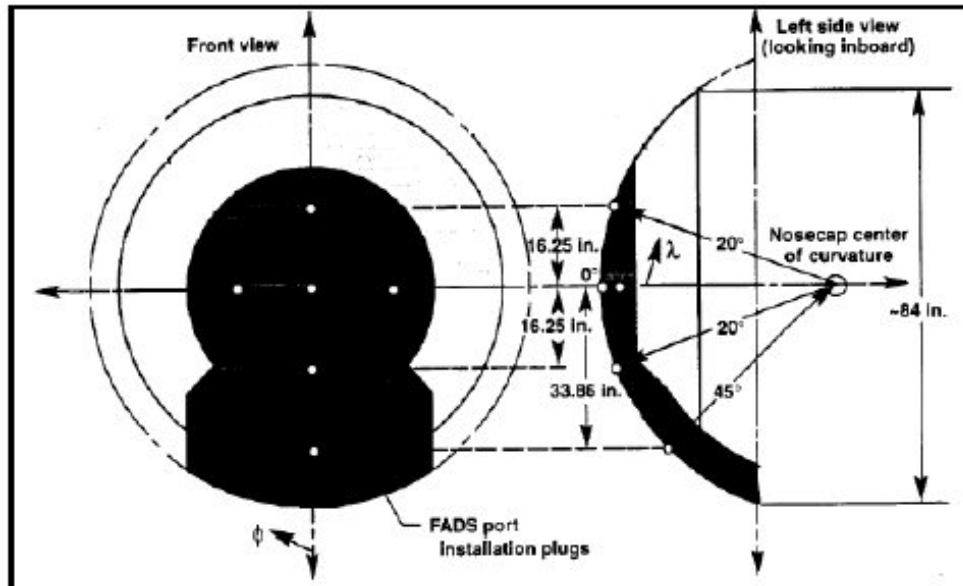
Le caratteristiche e gli obiettivi di questo progetto possono essere sintetiz-



**Figura 3.2:** Esempio di sistema FADS installato sullo Shuttle X-33.

zati come segue:

- La sonda presenta un numero di prese di pressione elevate poiché si



**Figura 3.3:** Disposizione delle prese di pressione del sistema FADS sull'X33.



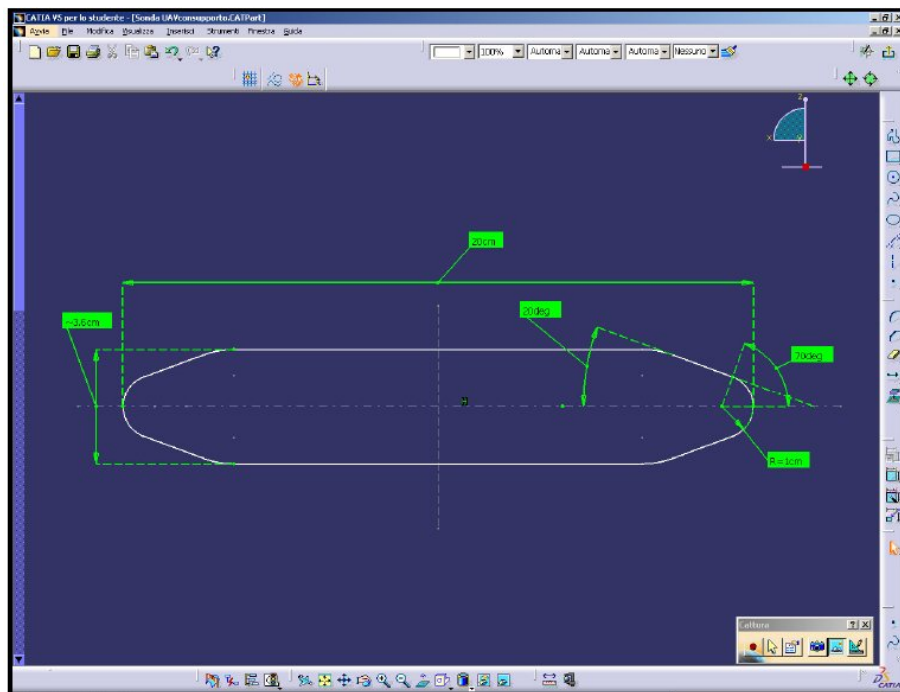
**Figura 3.4:** Velivolo non abitato dotato di apparecchiatura di monitoraggio del territorio.

ispira ai moderni sistemi FADS ma è dotata di un supporto che ne rende possibile l'installazione sotto la fusoliera per velivoli non abitati con sistemi di monitoraggio del territorio sul *nose* (vedi Fig. 3.4);

- Le dimensioni sono state ridotte il più possibile poiché si tratta di una sonda di tipo non intrusivo;
- Infine si è fatto in modo che la geometria della sonda non crei problemi di natura aerodinamica, elevata resistenza o punti di distacco del flusso.

Partendo da queste considerazioni è stata disegnata, con il *software* di modellazione tridimensionale *Catia*<sup>®</sup>, una geometria preliminare della sonda.

La Fig. 3.5 mostra il corpo principale dello strumento: la parte anteriore è



**Figura 3.5:** Vista dall'alto del corpo principale della sonda di pressione.

sostanzialmente costituita da una calotta sferica di raggio 1 cm per l'alloggiamento delle prese di pressione e da un corpo cilindrico di diametro 3,57 cm, raccordato alla calotta attraverso una superficie tronco conica di semiapertura 20°. Tale raccordo permette di non avere interferenze aerodinamiche ad alte incidenze e garantisce una superficie anteriore sufficientemente grande da posizionare tutte le prese di pressione necessarie, nonostante un raggio di dimensioni piuttosto ridotte.

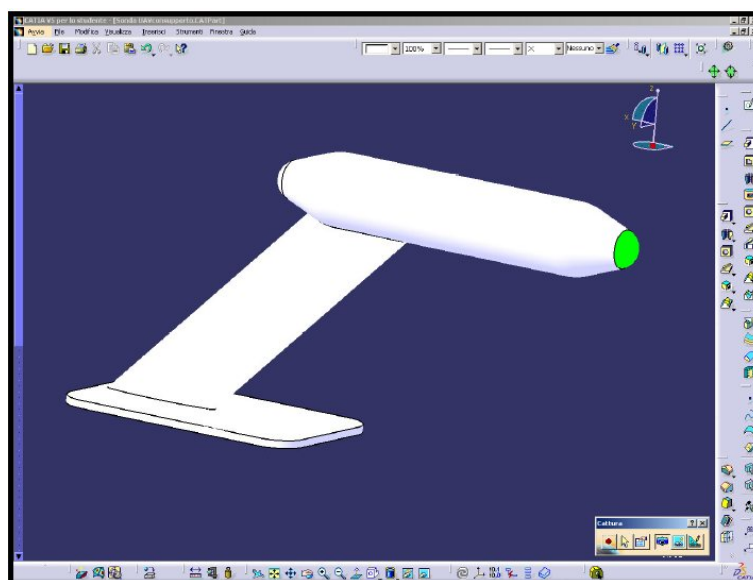
Per quanto riguarda la scelta delle dimensioni del cilindro questa è stata fatta in modo da rendere possibile l'inserimento dei cavi per la trasmissione dei segnali, dalle prese di misura fino all'unità dedicata alla loro elaborazione.

La parte terminale della sonda è stata disegnata invece in modo da minimizzare la resistenza aerodinamica: l'ingombro totale del sensore è dunque di circa 20 cm.

Il supporto è costituito da una piastra rettangolare per l'ancoraggio al velivolo e da un corpo verticale che ha il compito di distanziare la sonda dalla fusoliera.

Le caratteristiche dimensionali del supporto ed il suo posizionamento rispetto al corpo principale sono riportate dettagliatamente in<sup>[17]</sup>. Anche per quanto riguarda la progettazione del supporto si è cercato di minimizzare gli ingombri ed evitare problemi di natura aerodinamica.

In Fig. 3.6 è riportata la vista tridimensionale del modello completo, costituito dalla sonda e dal supporto d'installazione, che è stato utilizzato nella successiva analisi CFD.



**Figura 3.6:** Vista tridimensionale della sonda completa.

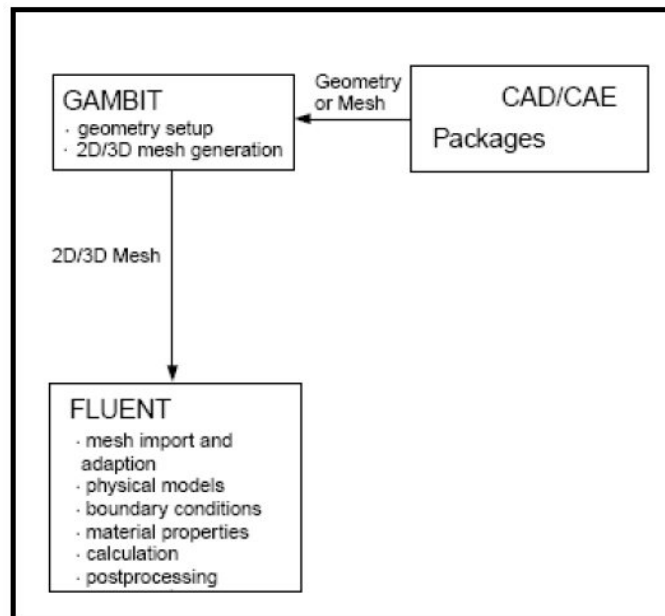
## 3.2 Realizzazione della mesh tramite il software Gambit

In Fig. 3.7 è riportato il flusso logico alla base del lavoro svolto: dopo aver realizzato la geometria con *Catia*<sup>®</sup>, è stata generata la *mesh*, o griglia di calcolo, in *Gambit* ed infine è stata effettuata l'analisi CFD, con il *software* commerciale *Fluent*.

Ogni passaggio verrà descritto in dettaglio nelle pagine che seguono.

La generazione della *mesh* e l'analisi fluidodinamica sono state eseguite su di un modello semplificato della sonda, mostrato in Fig. 3.8, rimandando alla fase di calibrazione in galleria del vento lo studio degli effetti di interferenza con il velivolo. La sonda è stata quindi analizzata in condizioni *stand alone*, trascurando il supporto verticale e metà corpo cilindrico in quanto le prese di misura sono posizionate solamente sulla calotta sferica che risulta di dimensioni ridotte rispetto all'intera sonda.

Una volta definito il modello, il *software Gambit* consente attraverso un'in-

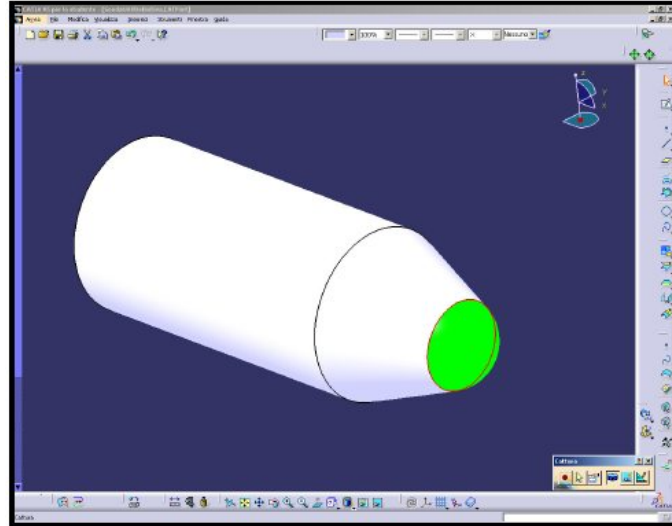


**Figura 3.7:** Flusso logico per la realizzazione della *mesh* e dell'analisi CFD.

terfaccia grafica (*Graphical User Interface*, GUI) di creare la *mesh*, settandone i relativi parametri ed assegnando le condizioni al contorno in modo semplice e facilmente applicabile a diverse tipologie di analisi.

Prima di importare il modello, occorre verificare che la geometria non presenti discontinuità o imperfezioni, ad esempio linee aperte che potrebbero generare errori durante la creazione della *mesh*.

Si osservi che quest'ultima deve esser realizzata sul flusso d'aria che investe la sonda: per questo motivo è necessario creare dei veri e propri *box* che contengono il modello. Durante il lavoro è emerso che contenitori di dimensioni confrontabili con quelle del modello possono dar luogo ad interferenze



**Figura 3.8:** Modello semplificato della sonda.

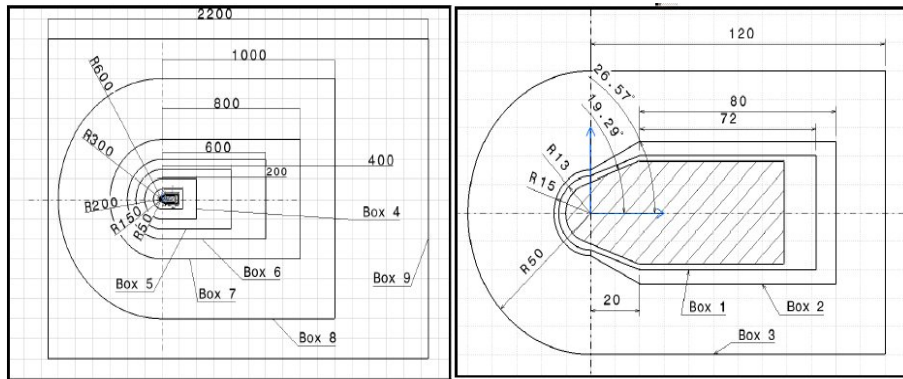
tra il flusso nei pressi di quest'ultimo e quello imperturbato sul contorno del contenitore più esterno; al contrario la creazione di *box* troppo grandi può portare ad un aumento dei tempi di calcolo: è stato quindi necessario trovare un compromesso tra queste due esigenze.

Inoltre la scelta delle dimensioni del *box* esterno è stata effettuata in modo tale che il flusso lontano dal modello risulti di tipo *free air* ovvero imperturbato.

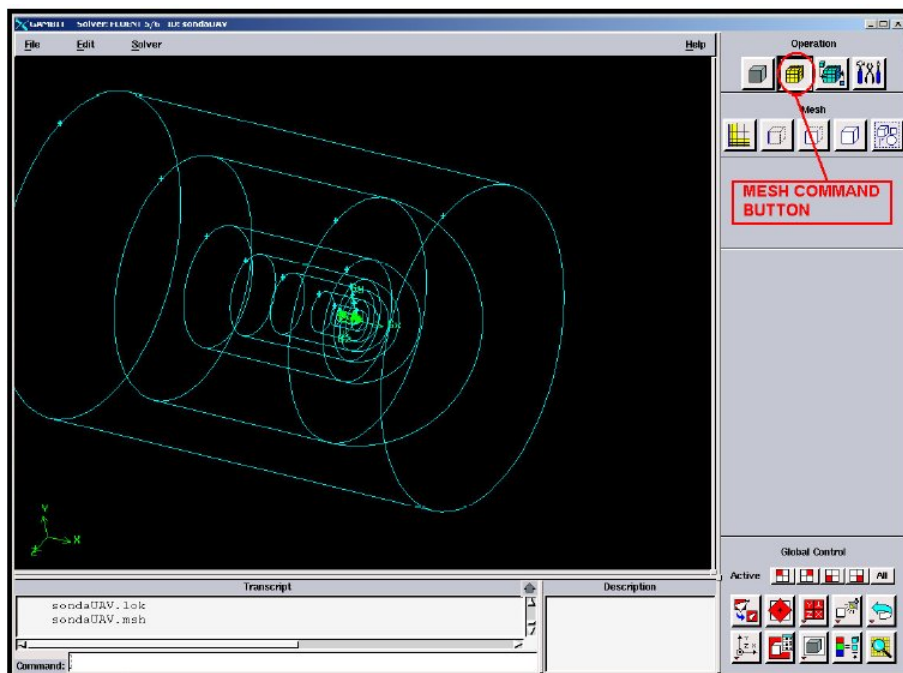
Le caratteristiche del *box* esterno sono riportate di seguito:

- a monte, 6 volte la lunghezza del modello;
- a valle, 15 volte la lunghezza del modello;
- diametro pari a 45 volte il diametro del modello.

La *mesh* presenta dimensioni ridotte in prossimità del modello, in modo da ottenere una buona precisione nei risultati, mentre allontanandoci da esso le dimensioni aumentano poiché non è più necessaria alcuna accuratezza di calcolo: questa scelta progettuale ha permesso di ottimizzare le risorse di calcolo riducendo i tempi delle singole simulazioni. Per realizzare una variazione graduale delle dimensioni della *mesh*, è stato necessario creare nove *box* intermedi, concentrici che, nelle vicinanze del modello, ne seguono la forma, mentre allontanandoci da esso si riducono a semplici cilindri. Il *software*



**Figura 3.9:** Dimensioni dei nove *box* realizzati intorno al modello.



**Figura 3.10:** Vista tridimensionale dei *box*.

dopo aver realizzato la *mesh* superficiale di dimensioni desiderate, da questa estrapola automaticamente i volumi con i quali riempire i *box*: questo meccanismo permette di controllare la crescita degli elementi di riempimento dal contorno del modello, dove il loro numero è piuttosto elevato, fino al *box*

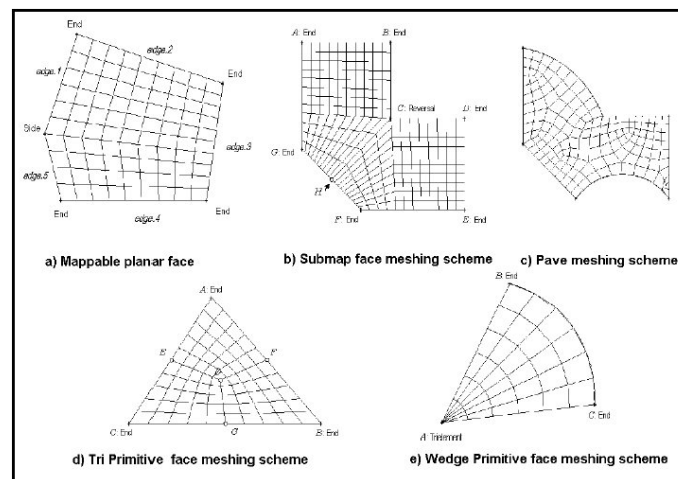


esterno. Dopo aver definito i *box*, si è passati alla realizzazione della *mesh*: l'interfaccia grafica di *Gambit* presenta una sezione dedicata al settaggio della griglia (*mesh*) di superficie e una per quella di volume, attraverso cui è possibile specificare la forma degli elementi ed il tipo di modello di generazione della griglia stessa. La compatibilità tra elementi e modelli è mostrata in Fig. 3.11 e Fig. 3.12.

Nel presente lavoro la *mesh* di superficie è stata realizzata con elementi di

Type	Elements		
	Quad	Tri	Quad/Tri
Map	X		X
Submap	X		
Pave	X	X	X
Tri Primitive	X		
Wedge Primitive			X

**Figura 3.11:** Applicabilità degli elementi di superficie ai relativi modelli di riempimento.



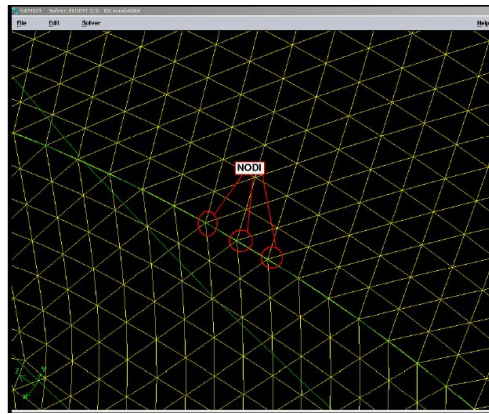
**Figura 3.12:** Modelli di riempimento.

forma triangolare su un modello di tipo *Pave*: questa scelta è dettata dal fatto che altri elementi avrebbero creato problemi sulla calotta sferica. Successivamente è stata definita la spaziatura tra i nodi: nel nostro caso il valore

è stato fornito direttamente tramite il comando *Interval Size*.

Come accennato in precedenza, per rendere il modello accurato e ridurre i tempi di simulazione, la spaziatura tra nodi (dunque la dimensione della griglia) aumenta gradualmente a partire dalla calotta sferica dove sono posizionate le prese (0.03cm), fino al corpo cilindrico (0.07cm), poiché quest'ultimo parte non interessa ai fini dell'analisi. Si osservi che, come mostrato in Fig. 3.13, sulle linee di contorno comuni a due o più superfici, il *software* mantiene i nodi creati con la prima spaziatura (0.03cm) e da questi genera gradualmente la seconda (0.07cm).

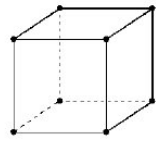
Per quanto riguarda elementi e modelli di riempimento della *mesh* di volu-



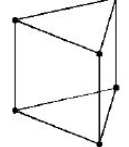
**Figura 3.13:** Nodi posizionati sulla linea di contorno tra due superfici.

me, anche in questo caso occorre tener presente le possibili applicazioni (vedi Fig. 3.14). La spaziatura tra i nodi, sui *box* più vicini al modello, è di circa 0.1cm sulla calotta ed aumenta fino a 0.12cm sul corpo cilindrico, mentre si arriva a circa 15cm sul box più esterno. Per quanto riguarda i riempimenti, sono stati utilizzati elementi di forma tetraedrica abbinati al modello *Tgrid*, che dove impossibilitato ad utilizzare questa forma, sfrutta elementi piramidali, esaedrici o d'angolo. La Fig. 3.15 mostra l'incremento delle dimensioni degli elementi di riempimento dalla calotta sferica verso l'esterno: questo meccanismo automatico consente di controllarne zona per zona la crescita, ridurre i tempi di calcolo ed affinare la precisione dei risultati solo sulla calotta sferica d'interesse.

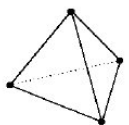
	Elements Option		
Type Option	Hex	Hex/Wedge	Tet/Hybrid
Map	X		
Submap	X		
Tet Primitive	X		
Cooper	X	X	
TGrid			X
Stairstep	X		



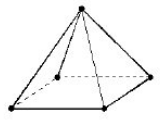
Hexaedral mesh element



Wedge mesh element

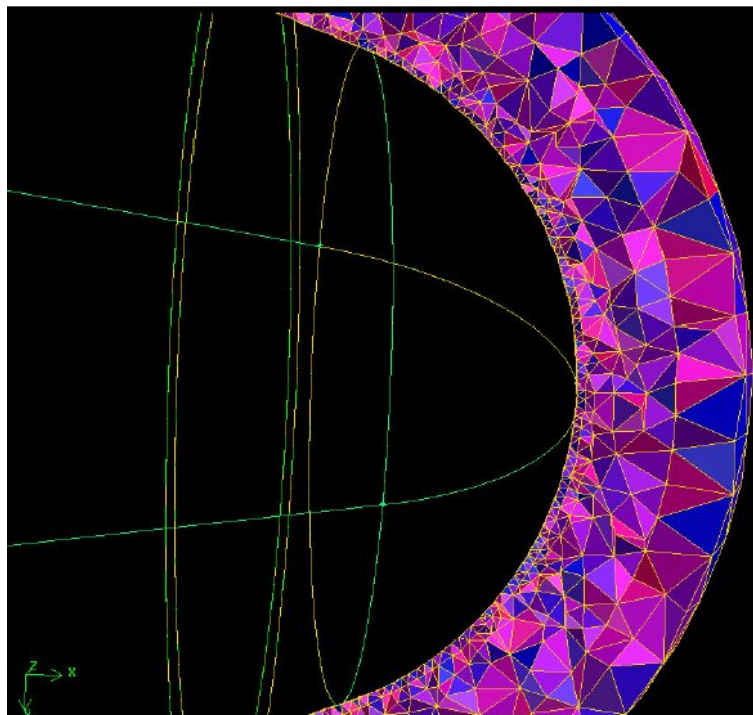


Tetrahedral mesh element



Pyramidal mesh element

**Figura 3.14:** Applicabilità degli elementi di volume ai relativi modelli di riempimento.



**Figura 3.15:** Variazione della dimensione degli elementi di volume in prossimità della calotta sferica.

### 3.3 Analisi CFD e generazione del database aerodinamico

Terminata la *mesh*, il modello viene esportato in *Fluent*, *software* commerciale che permette di simulare attraverso un'analisi di fluidodinamica computazionale (*Computation Fluid Dynamic*, CFD) il comportamento dei fluidi in complicati problemi fisici. Tale simulazione consiste nell'elaborazione numerica di sofisticati modelli matematici che descrivono l'evoluzione temporale dei parametri fondamentali del fluido quali pressione, velocità, temperatura e densità.

In quest'ultimi anni, grazie soprattutto all'enorme sviluppo delle capacità dei calcolatori, la fluidodinamica numerica ha avuto un'evoluzione tale da renderla strumento indispensabile per la progettazione in molti settori dell'ingegneria. Occorre sottolineare però che, nonostante l'uso di equazioni semplificate e *computer* veloci si riesce ad ottenere soltanto soluzioni approssimative.

L'analisi CFD è risultata uno strumento utile per affinare la geometria scelta, prima della sperimentazione e del progetto di dettaglio: ciò comporta l'ottimizzazione dell'intera fase progettuale, riducendone tempi e costi. Inoltre tale analisi ha consentito di ottimizzare il posizionamento delle prese di misura ed ha portato alla definizione di un *database* preliminare di pressioni locali su cui andare in seguito ad allenare le reti per la ricostruzione dei parametri di volo. Si osservi che la CFD fornisce, a differenza delle simulazioni in galleria del vento, risultati indipendenti dal fattore di scala.

Il moto del fluido è descritto dalle equazioni di Navier-Stokes, ovvero dall'equazione di continuità della massa (in coordinate cartesiane):

$$\frac{Ps_{local_i}}{P_{sa}} = \left(1 + \frac{\gamma}{2} \cdot Cps_{local_i} \cdot M_{\infty}^2\right) \quad (3.1)$$

dall'equazione di conservazione della quantità di moto:

$$\frac{Ps_{local_i}}{P_{sa}} = \left(1 + \frac{\gamma}{2} \cdot Cps_{local_i} \cdot M_{\infty}^2\right) \quad (3.2)$$

e dall'equazione di conservazione dell'energia.

$$\frac{Ps_{local_i}}{P_{sa}} = \left(1 + \frac{\gamma}{2} \cdot Cps_{local_i} \cdot M_{\infty}^2\right) \quad (3.3)$$

Nelle Eq. (3.1), (3.2), (3.3)  $\rho$  indica la densità del fluido,  $\mu$  la viscosità dinamica,  $C_p$  il calore specifico a pressione costante,  $T$  e  $P$  la temperatura e la pressione del fluido,  $q$  il calore scambiato per unità di massa,  $G$  forze per unità di volume,  $V$  velocità del flusso ed infine  $\tau_{ij}$  le tensioni tangenziali di

deformazione.

Le Eq. (3.1), (3.2), (3.3) rappresentano un sistema di equazioni differenziali alle derivate parziali, non lineare che, a causa della notevole complessità, non può esser risolto in generale in forma chiusa. Quindi occorre fare ipotesi semplificative sul tipo di fluido e di flusso: ad esempio nel caso per i fluidi non viscosi possono esser risolte le equazioni di Eulero, mentre per i flussi potenziali si passa alle equazioni di Laplace.

La risoluzione numerica delle equazioni complete di Navier-Stokes avviene utilizzando essenzialmente tre metodologie, descritte ampiamente in<sup>[18]</sup> :

- *Direct Numerical Simulation*, DNS;
- *Large Eddy Simulation*, LES;
- *Reynolds Averaged Navier-Stokes equation*, RANS.

Nel presente lavoro è stato utilizzato l'approccio RANS, attualmente il più diffuso per la risoluzione di problemi d'interesse ingegneristico, poiché rispetto ad esempio alla DNS riduce notevolmente i costi computazionali. Le equazioni di Navier-Stokes vengono risolte considerando le variabili fisiche dipendenti dal tempo come somma di due contributi: il valor medio in un intervallo di tempo caratteristico e la perturbazione rispetto a tale valor medio. La tecnica RANS risulta esser una metodologia approssimata di soluzione delle equazioni poiché non valuta l'effetto locale della turbolenza sulle grandezze medie.

Va infine sottolineato che le RANS forniscono risultati soddisfacenti quanto più il modello di turbolenza riesce ad approssimare la fisica del problema in oggetto.

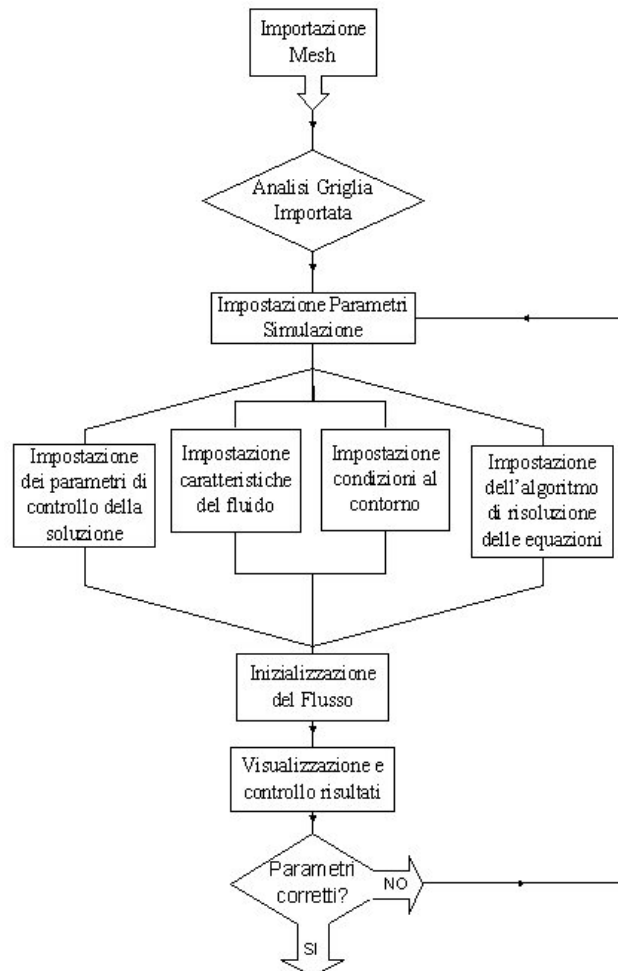
Il dominio fluido viene discretizzato dal calcolatore col metodo dei volumi finiti, ovvero viene suddiviso in tante celle elementari di piccole dimensioni (elementi) che formano la griglia di calcolo (*mesh*). A quest'ultima si applica il metodo RANS in modo iterativo, fino a che non viene raggiunto il grado di accuratezza fissato a priori sulle misure di pressione locali.

### 3.3.1 Impostazione dei parametri di simulazione

L'impostazione delle analisi CFD segue il flusso logico in Fig. 3.16.

Dopo aver importato la griglia o dominio fisico in *Fluent*, è stato necessario controllare la geometria per evitare problemi legati ad eventuali contorni aperti e scegliere la tipologia di soluzione tramite dei menù di interfaccia grafica.

In particolare nel presente lavoro sono state effettuate le seguenti scelte:



**Figura 3.16:** Visualizzazione del flusso logico per effettuare le simulazioni CFD.

- le analisi effettuate sono di tipo stazionario (*Steady*) in modo da eliminare la dipendenza dei risultati dal tempo;
- le pressioni sono calcolate come media aritmetica nel centro della singola griglia;
- la presenza della turbolenza è stata introdotta in modo semplificato utilizzando il modello  $\kappa$ - $\varepsilon$ , che risulta compatibile con la tecnica RANS. Questo modello introduce due equazioni aggiuntive nelle variabili energia cinetica di turbolenza e rateo di dissipazione dell'energia stessa. In

particolare sulle pareti, *Fluent* permette di impostare il *Non Equilibrium Wall* considerando nulle le variazioni dei parametri legate alla turbolenza.

La scelta delle condizioni al contorno viene effettuata, sia per quanto riguarda le superfici che i volumi, attraverso il comando *Zone Command*. In particolare, i volumi vengono trattati come *FLUID*, mentre per le superfici sono possibili le seguenti impostazioni:

- *Wall*: tale impostazione viene adottata per le superfici della sonda dove si richiede velocità nulla ovvero non scorrimento alla parete;
- *Velocity Inlet*: tale impostazione riguarda le superfici del *box* esterno. Su di esse la velocità è costantemente uguale alla velocità del flusso imperturbato poiché la sonda viene studiata in condizioni *stand alone*;
- *Interior*: impostazione adottata sulle superfici dei *box* più interni, poiché i parametri che caratterizzano il flusso devono restare costanti.

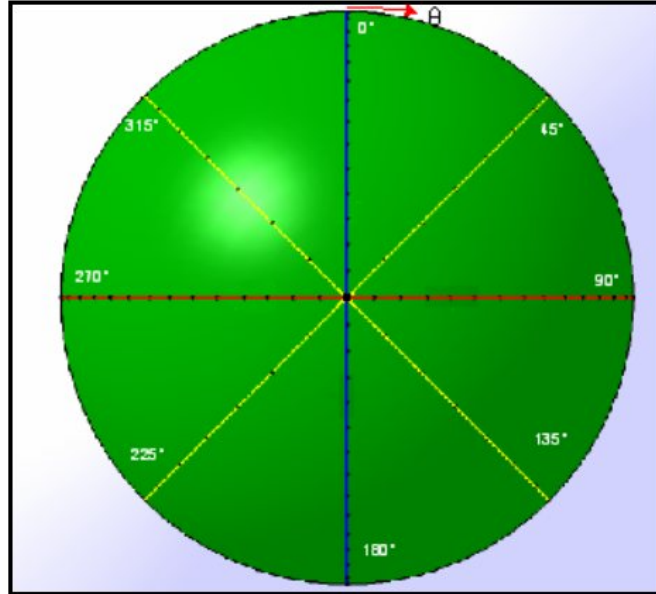
Il controllo della soluzione è stato eseguito sia sulle equazioni del flusso sia sulle due equazioni di turbolenza. Definiti degli opportuni volumi di controllo, il valore della generica grandezza lungo il bordo è ricavato tramite interpolazione, con riferimento al centro della cella (schema *upwind*): questa impostazione determina nella soluzione un'accuratezza del primo ordine. Per alcune grandezze, quali momento, energia cinetica e rapporto di dissipazione dell'energia si è utilizzato invece il *second order upwind*, che determina un'accuratezza del secondo ordine.

Dopo aver impostato i parametri di simulazione, si è passati alla definizione dei punti di controllo, ovvero delle prese di misura delle pressioni locali sulla superficie del modello. Come mostrato in Fig. 3.17, sono stati scelti quattro archi di circonferenza e su di essi sono stati individuati 85 punti: sugli archi in rosso e blu i punti sono stati posizionati ad intervalli di  $5^\circ$ , sui restanti archi invece, essendo questi di minor interesse per il nostro studio, l'intervallo è di  $10^\circ$ .

Un'ampia campagna di simulazioni ha portato alla definizione in tutto l'inviluppo di volo desiderato, di un *database* preliminare di pressioni locali. Le simulazioni sono state effettuate per velocità appartenenti all'intervallo  $20\text{m/s} \leq v \leq 70\text{m/s}$ , con passo di  $10\text{m/s}$ , per angoli di incidenza compresi in  $-5^\circ \leq \alpha \leq 15^\circ$  e derapata tra  $-15^\circ \leq \beta \leq 15^\circ$ , entrambe con passo di  $5^\circ$ .

### 3.3.2 Visualizzazione ed analisi dei risultati

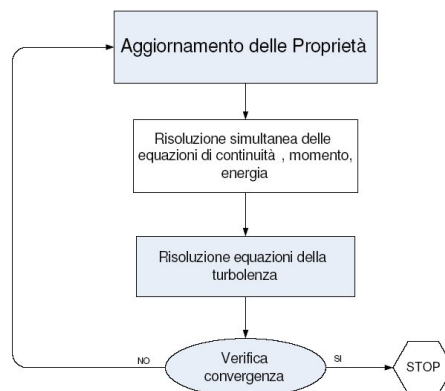
Dal punto di vista computazionale, *Fluent* risolve, in funzione delle condizioni iniziali scelte, dapprima le equazioni di continuità, di momento ed energia



**Figura 3.17:** Posizione delle prese di misura sugli archi principali della sonda.

e successivamente quelle relative alla turbolenza: il *software* confronta i risultati ottenuti con opportuni valori di riferimento in modo da valutarne la convergenza (vedi flusso logico in Fig. 3.18).

In particolare in Fig. 3.19 è riportato l'andamento dei residui sulle tre com-



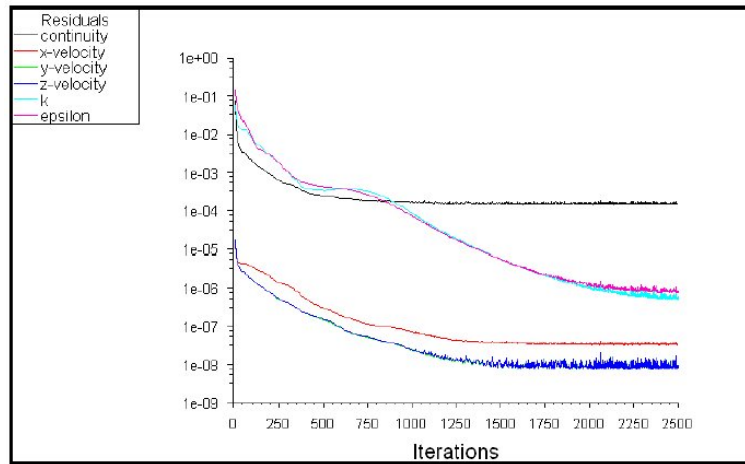
**Figura 3.18:** Flusso logico per verificare la convergenza della soluzione.

ponenti  $x, y, z$  della velocità, su  $k$  ed  $\epsilon$ , in funzione del numero di iterazioni.



Quest'analisi serve a stabilire il numero minimo di iterazioni (2500) necessarie per portare a convergenza la soluzione.

Si osservi che l'andamento dei residui non dipende dalle caratteristiche della simulazione o dai valori dei angoli  $\alpha$ ,  $\beta$ , poiché i residui sono parametri legati a problemi di convergenza di equazioni fluidodinamiche. La Fig. 3.20 mostra,



**Figura 3.19:** Andamento dei residui in funzione del numero delle iterazioni.

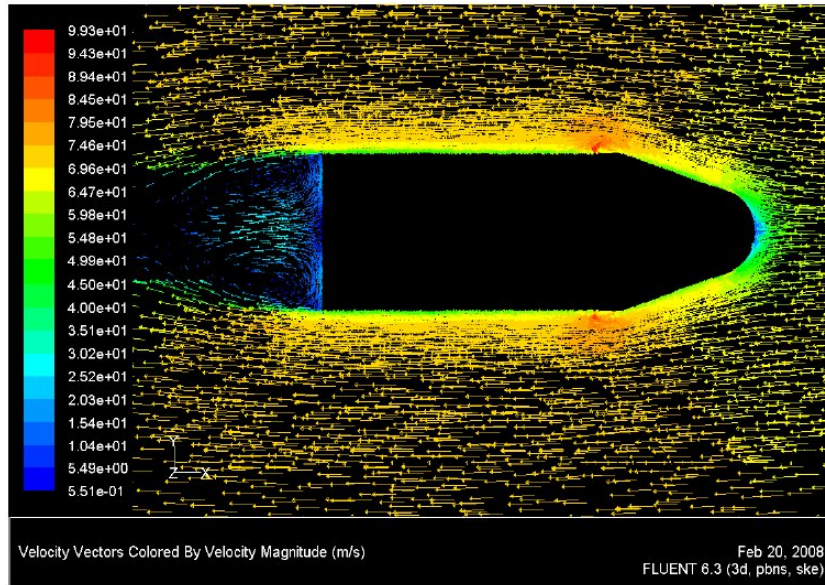
per una simulazione ad  $\alpha$ ,  $\beta$  nulli e  $v = 70m/s$ , l'andamento dei vettori velocità sul piano  $xy$  del modello: ciò permette di apprezzare le linee di corrente, i punti di ristagno, i vortici alla base del modello e verificare eventuali errori nell'impostazione della simulazione.

I dati raccolti nella campagna di simulazioni sono stati analizzati tramite il software *Matlab*<sup>®</sup> e gestiti sotto forma di matrici. La prima colonna contiene un numero da 0 a 84 che identifica il punto di controllo preso in considerazione di volta in volta. Fissato il punto e muovendoci lungo la riga della matrice troviamo le coordinate  $(x, y, z)$  del punto ed infine la pressione locale come variazione  $\Delta P$  rispetto al valore al *sea level*, a  $15^\circ C$ .

In questo modo è stata realizzata una griglia di pressioni locali in funzione di  $\alpha$ ,  $\beta$  e  $v$ .

Per ridurre il numero di simulazioni da effettuare, il valore della pressione è stato spesso ricavato sia sfruttando la simmetria geometrica del problema sia tramite interpolazione tra due valori precedentemente simulati.

Infatti, come mostrato in Fig. 3.21 e Fig. 3.22, effettuando le simulazioni ad  $\alpha = 0^\circ, 5^\circ, 10^\circ, 15^\circ$  e  $\beta = 0^\circ$  ad una velocità di  $20m/s$  e di  $70m/s$ , si può notare che c'è una simmetria tra i dati ottenuti al variare di  $\alpha$ , fissato  $\beta$  e  $v$ . Analogamente fissando questa volta gli angoli e facendo variare solo la



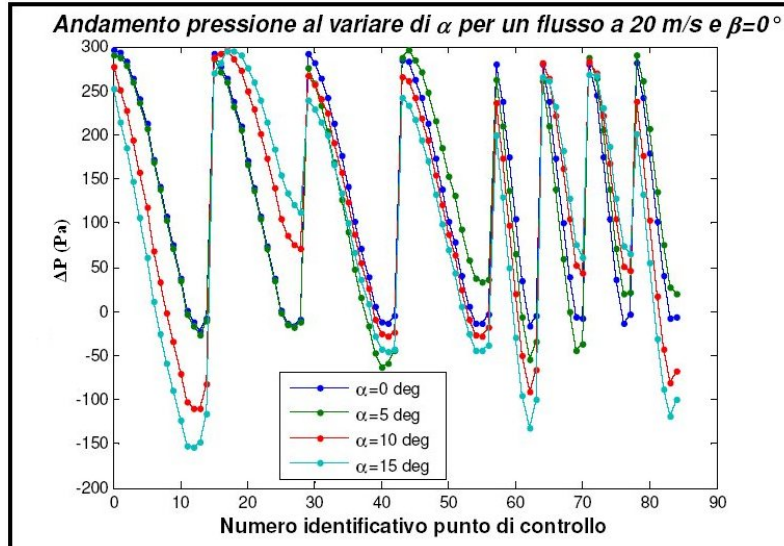
**Figura 3.20:** Visualizzazione dei vettori velocità sul piano  $xy$  per una simulazione ad  $\alpha, \beta$  nulli e  $v = 70\text{m/s}$ .

velocità  $v$  (vedi Fig. 3.23 e Fig. 3.24). Queste relazioni consentono di ricavare alcuni dati mancanti tramite interpolazione.

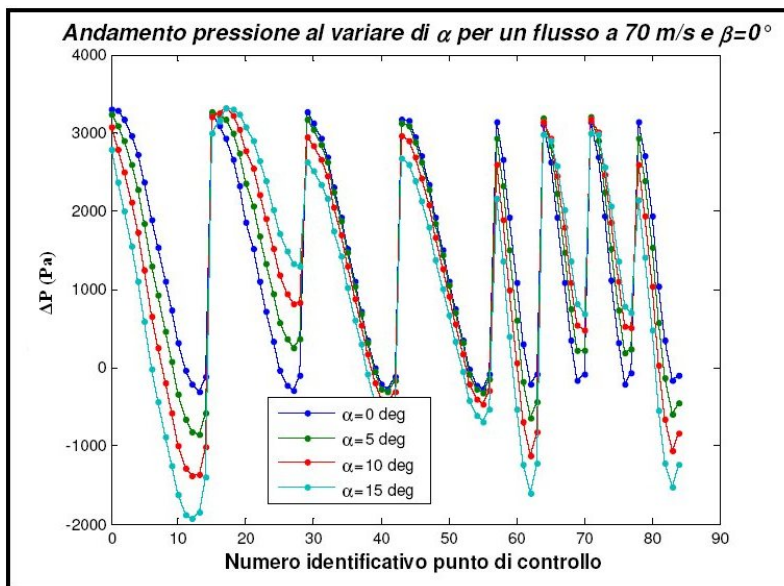
Inoltre, analizzando il *database* sono state riscontrate delle simmetrie nei valori di pressione, dovute alla simmetria geometrica del problema stesso: ad esempio, fissando l'angolo  $\alpha$ , i dati a  $\beta$  negativi possono esser ricavati semplicemente ribaltando intorno all'asse  $z$  quelli ottenuti a  $\beta$  positivi. La Fig. 3.25 mostra ad esempio l'andamento della  $\Delta P$  in funzione del punto di controllo, per  $\alpha = 5^\circ$ ,  $\beta = \pm 10^\circ$  e  $v = 20\text{m/s}$ . Analogamente, fissando  $\beta$ , i dati ad  $\alpha$  negativi sono ottenuti ribaltando intorno all'asse  $y$  quelli ad  $\alpha$  positivi.

La griglia di dati alle velocità estreme (20 m/s e 70 m/s) è stata completata sfruttando le relazioni di simmetria e le interpolazioni. In Fig. 3.26 e Fig. 3.27 sono raffigurate le griglie di valori a 20 m/s e 70 m/s: in rosso sono indicati i punti in cui la simulazione è stata realmente eseguita attraverso *Fluent*®, in blu sono rappresentati i dati ricavati per simmetria ed in nero i valori interpolati. Inoltre in quest'ultimo caso, sono state eseguite delle simulazioni di controllo per avere una stima dell'errore commesso in fase di interpolazione.

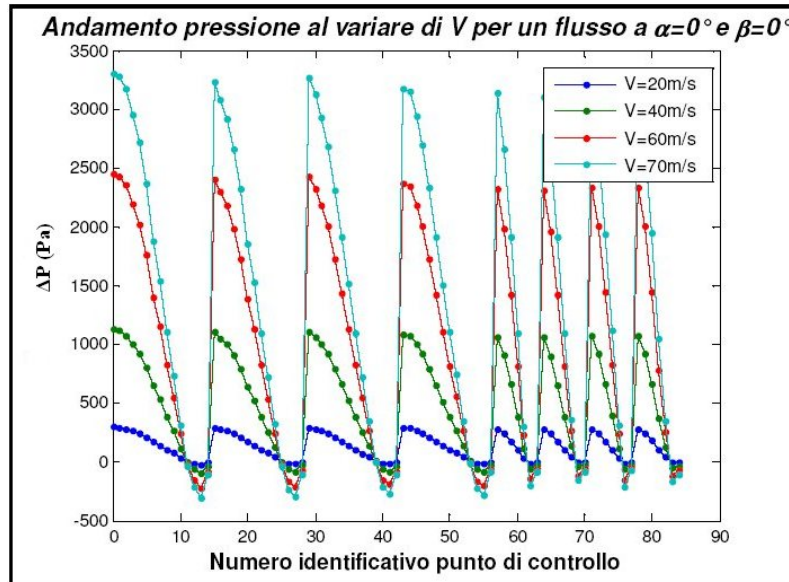
Dopo aver completato le griglie a 20 m/s e 70 m/s, sono state effettuate tutte le simulazioni ad  $\alpha = 0^\circ$  e  $\beta = 0^\circ$ , ad intervalli di velocità di 5 m/s, in modo da avere una stima della variazione di pressione in funzione della velocità



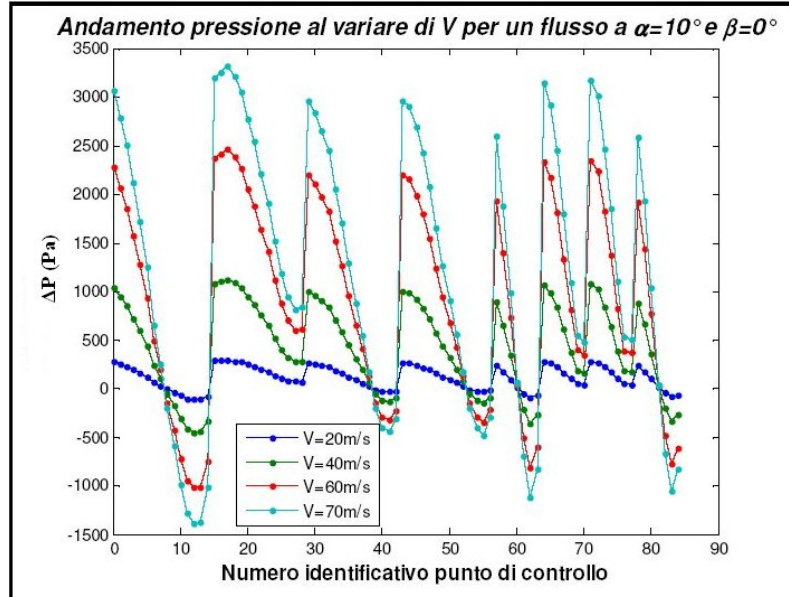
**Figura 3.21:** Andamento della  $\Delta P$  in funzione del punto di controllo, al variare di  $\alpha$  per  $\beta = 0^\circ$  e  $v = 20\text{m/s}$ .



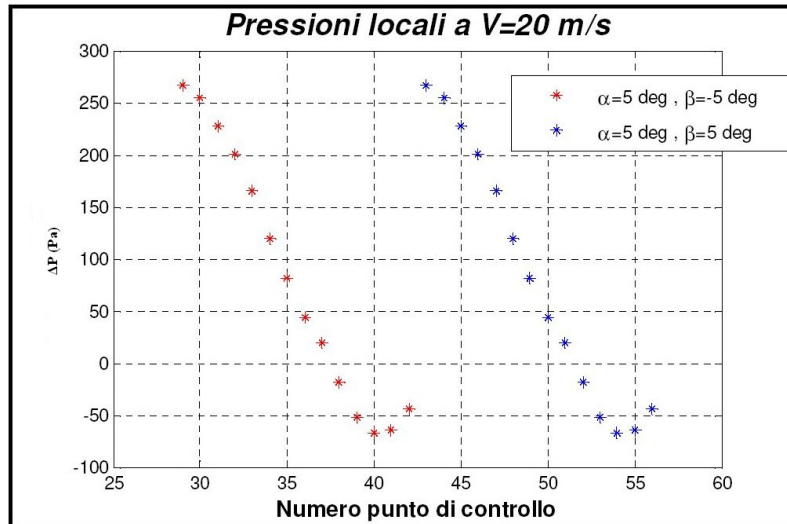
**Figura 3.22:** Andamento della  $\Delta P$  in funzione del punto di controllo, al variare di  $\alpha$  per  $\beta = 0^\circ$  e  $v = 70\text{m/s}$ .



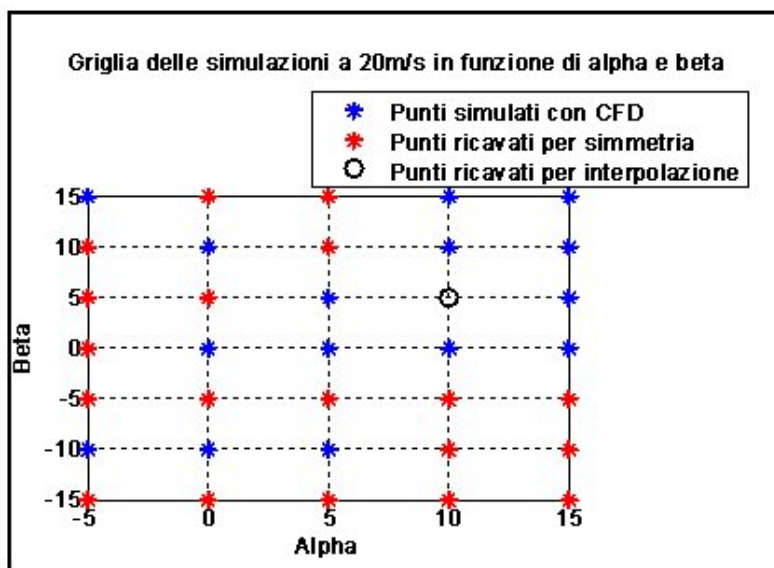
**Figura 3.23:** Andamento della  $\Delta P$  in funzione della velocità per  $\alpha$  e  $\beta$  nulli.



**Figura 3.24:** Andamento della  $\Delta P$  in funzione della velocità per  $\alpha = 10^\circ$  e  $\beta = 0^\circ$ .

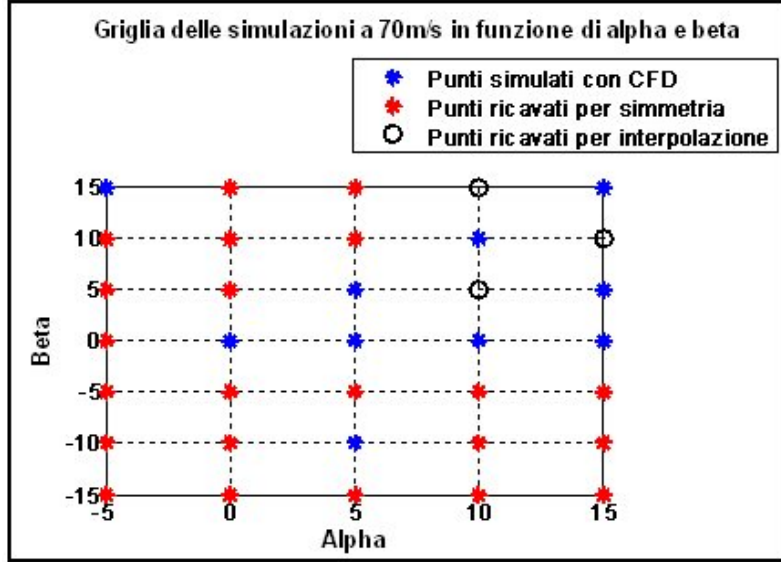


**Figura 3.25:** Visualizzazione delle simmetrie geometriche: andamento della  $\Delta P$  in funzione del punto di controllo per  $v = 20\text{ m/s}$ .



**Figura 3.26:** Griglia delle simulazioni effettuate a  $v = 20\text{ m/s}$  in funzione di  $\alpha$  e  $\beta$  misurati in gradi.

stessa. poiché tale relazione non è lineare, l'interpolazione dei dati al variare della velocità ha richiesto l'uso di opportuni pesi, calcolati attraverso i dati



**Figura 3.27:** Griglia delle simulazioni effettuate a  $v = 70m/s$  in funzione di  $\alpha$  e  $\beta$  misurati in gradi.

provenienti dalle simulazioni ad angoli d'incidenza e di derapata nulli, nel punto di controllo numero 0 (punto centrale sulla calotta sferica). Il peso da applicare al generico punto di controllo  $n$ , ad  $\alpha$  e  $\beta$  generici, viene ricavato attraverso la relazione:

$$\Delta P = (f_p \cdot D) + \Delta P(20, \alpha, \beta, n) \quad (3.4)$$

dove:

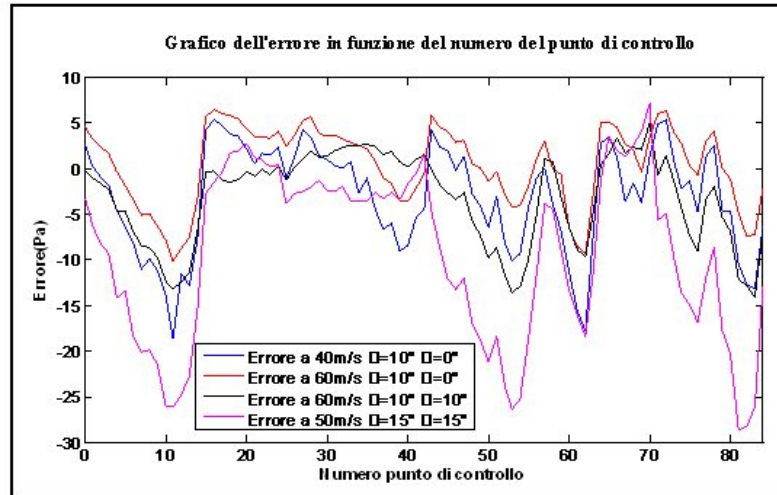
$$f_p = \frac{\Delta P(v, 0, 0, 0) - \Delta P(20, 0, 0, 0)}{\Delta P(70, 0, 0, 0) - \Delta P(20, 0, 0, 0)} \quad (3.5)$$

e con:

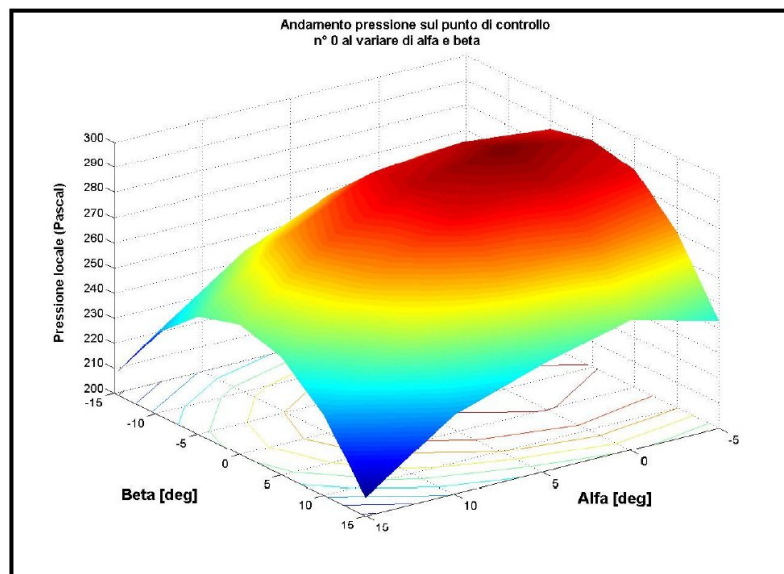
$$D = \Delta P(70, \alpha, \beta, n) - \Delta P(20, \alpha, \beta, n) \quad (3.6)$$

Sono state effettuate delle simulazioni di controllo per analizzare l'errore commesso durante le interpolazioni (Fig. 3.28). Successivamente per ogni punto di controllo, è stata realizzata una matrice tridimensionale di  $\Delta P$ , in funzione del valore dell'angolo di incidenza, di derapata e della velocità: la Fig. 3.29 mostra la superficie che rappresenta l'andamento della  $\Delta P$  nel punto di controllo centrale, per  $v = 20m/s$ .





**Figura 3.28:** Andamento dell'errore commesso durante le interpolazioni in funzione del punto di controllo.



**Figura 3.29:** Visualizzazione tridimensionale della  $\Delta P$  relativa al punto di controllo centrale, in funzione di  $\alpha$  e  $\beta$ , per  $v = 20\text{m/s}$ .

### 3.4 Tecniche di intelligenza artificiale per la ricostruzione dei dati aria

Le reti neurali artificiali, (*neural networks*, NNs), sono nate per riprodurre attività tipiche del cervello umano (la percezione di immagini, il riconoscimento di forme, la comprensione del linguaggio, il coordinamento senso-motorio, ...) e possono esser definite come *reti di elementi interconnessi in parallelo in larga misura e la cui organizzazione gerarchica è progettata in modo da interagire con gli oggetti del mondo reale in modo simile al sistema nervoso biologico*: dunque per poter creare una rete neurale artificiale occorre studiare il funzionamento del nostro cervello.

Nelle pagine che seguono è riportata una breve analogia tra il funzionamento del sistema nervoso umano e le reti neurali utilizzate per la ricostruzione dei dati aria. Inoltre vengono descritte, oltre alle caratteristiche strutturali, le metodologie di allenamento, di test, di validazione ed i risultati ottenuti.

#### 3.4.1 La metafora neurobiologica

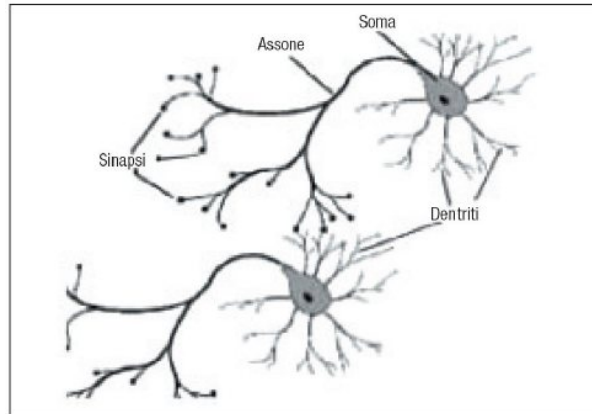
Il sistema nervoso umano è costituito da miliardi di cellule nervose dette *neuroni* (vedi Fig. 3.30), sedi di processi elettrochimici responsabili della generazione di campi elettromagnetici. Il corpo principale (*soma*) presenta due prolungamenti ramificati: i *dendridi*, attraverso cui il neurone scambia segnali elettrici con gli altri neuroni, e l'*assone*, la cui lunghezza varia da 1cm a qualche metro. L'estremità dell'assone è a sua volta ramificata per permettere la trasmissione dei segnali ai dendridi di altri neuroni. Tra un terminale di un assone e la cellula ricevente esiste uno spazio: i segnali superano questo spazio per mezzo di sostanze chimiche detti neurotrasmettitori. Il punto di connessione tra terminale e dendride è chiamato *sinapsi*.

Un neurone si attiva, ovvero trasmette impulsi elettrici lungo il suo assone, quando si verifica una differenza di potenziale tra interno ed esterno della cellula: quest'impulso determina una liberazione di un neurotrasmettitore dai terminali dell'assone.

Il cervello umano, calcolatore complesso e non lineare, pur essendo costituito da elementi di elaborazione semplici (neuroni) è in grado di eseguire computazioni complesse (riconoscimento, percezione e controllo del movimento, ...): occorre sottolineare comunque che i neuroni biologici sono 5 volte più lenti dei componenti elettronici.

Le capacità cognitive, legate all'elaborazione dei segnali elettrici presenti nei neuroni, sono influenzate dalla vita dell'individuo e dalle esperienze fatte. Si è verificato sperimentalmente che le funzioni cognitive, che risiedono in particolari zone del cervello, possono esser perdute in seguito alla rottura dei legami sinaptici ed eventualmente recuperate in seguito a nuovi processi di





**Figura 3.30:** Neurone biologico: visualizzazione dei costituenti.

apprendimento capaci di generare nuovi *pattern* di interconnessione sinaptica.

Da ciò si evince che il cervello umano è in grado di modificare le connessioni tra i neuroni in base all'esperienza acquisita ovvero è in grado di imparare. Inoltre nel cervello non esiste un controllo centralizzato: il cervello è *fault-tolerant* ovvero se un neurone o una delle sue connessioni sono danneggiate, è comunque garantito il funzionamento, pur con prestazioni degradate.

La computazione neurale artificiale cerca dunque di modellare e simulare le funzioni di base dei sistemi biologici: numero elevato di elementi semplici, struttura distribuita, funzionamento in parallelo, capacità di apprendere e generalizzare ovvero di produrre uscite in corrispondenza di ingressi non incontrati durante l'addestramento. La computazione neurale si contrappone dunque alla filosofia in cui un singolo processore accentra tutta la capacità computazionale ed esegue le operazioni in sequenza programmata.

Da un punto di vista storico come si evince dalla Fig. 3.31 il lavoro di *McCulloch e Pitts* è la prima analisi completa e formale che evidenzia come semplici unità, con sinapsi eccitatorie ed inibitorie dotate di apposita soglia, possano esser in grado di rappresentare proposizioni complesse nell'ambito di un processo collettivo. Tale lavoro è risultato rilevante soprattutto per gli sviluppi nel settore dei calcolatori piuttosto che nelle scienze cognitive.

Capire ed investigare i segreti della mente umana osservando l'attività cerebrale rappresenta sicuramente una sfida affascinante ma il *reverse engineering* ovvero la capacità di dedurre regole da esempi è un problema molto complesso.

<i>Cronologia delle Reti Neurali</i>	
<b>I era</b>	<b>Eventi significativi</b>
1943	McCulloch and Pitts, formalizzazione del neurone artificiale
1949	D.Hebb e l'apprendimento per auto-organizzazione
1956	'Dartmouth Summer Research Project on AI' con Minsky, Mccarty, Rochester, Shannon
1960	Widrow: ADALINE
1962	Il perceptron di Rosenblatt
1969	'Perceptrons', Minsky & Papert
70s	Periodo 'buio': degni di nota gli associatori di Anderson, i modelli per apprendimento senza supervisione di Kohonen, gli studi di Grossberg
<b>II era</b>	<b>Eventi significativi</b>
1982	Reti di Hopfield: memorie associative e risoluzione di problemi
1986	PDP e diffusione di Backpropagation
1987	La prima conferenza significativa dell' <i>IEEE</i> a SanDiego
1989	I chip neurali si affacciano sul mercato: <i>Analog VLSI and Neural Systems</i>
1990	J.Pollack e le reti neurali che elaborano strutture dati
1994	Prima Conferenza Mondiale sull'Intelligenza Computazionale (Orlando)
1994	Nasce il progetto NeuroCOLT <i>Computational Learning Theory</i>
2001	L' <i>IEEE</i> approva la creazione della <i>Neural Networks Society</i>

**Figura 3.31:** Eventi significativi per lo sviluppo di tecniche neurali.

### 3.4.2 Campi d'applicazione e possibili problematiche

Tipicamente le NN trovano applicazione nell'individuazione, in sistemi complessi, dei legami tra ingressi ed uscite nei casi in cui i dati a disposizione sono numerosi. Tenendo presente che le reti neurali non sono adatte a risolvere qualunque tipo di problema, di seguito viene riportato un elenco delle applicazioni più frequenti:

- nell'ambito di sistemi intelligenti basati su segnali d'ingresso di natura sensoriale: sistemi di riconoscimento della voce; sistemi di elaborazione dei suoni; rivelazione, verifica ed identificazione di caratteri (impronte digitali, codici a barre, firme, ...); riconoscimento di odori tramite sistemi olfattivi artificiali; elaborazione di immagini; sistemi di navigazione e guida;
- sistemi esperti;
- sistemi di controllo adattivi (controllo di processi o apparecchiature);
- robot autonomi.

Occorre sottolineare che in fase di progettazione di una rete neurale possono presentarsi sostanzialmente tre tipologie di problemi:

- relativamente all'**architettura della rete**: una rete è per natura adatta a risolvere un particolare problema. Occorrerebbe dunque utilizzare sempre reti capaci di risolvere il problema in esame: se non ci sono reti con questa caratteristica la soluzione sarà sicuramente affetta da un errore;
- relativamente all'**insieme di apprendimento**: la rete acquisisce esperienza dagli esempi che costituiscono l'insieme di apprendimento. Quindi è possibile immaginare una rete come un meccanismo di generazione basato sull'imitazione del cervello umano. La valutazione delle informazioni da inserire nell'insieme di apprendimento costituisce un problema ancora aperto;
- relativamente all'**algoritmo di apprendimento**: quest'algoritmo ha il compito di modificare l'architettura della rete ed i suoi parametri, in funzione delle informazioni contenute nell'insieme di apprendimento. Si tratta di un processo iterativo: è ancora aperto il problema della convergenza di tale algoritmo.

### 3.4.3 Il principio di Occam

La caratteristica che più interessa imitare nelle reti neurali è la capacità del nostro cervello di acquisire esperienza dai vari esempi concreti. E' chiaro che l'imitazione può esser guidata dal buon senso, che a differenza del semplice immagazzinamento di dati e dunque memorizzazione di regole esistenti, consente di astrarre una regola di generalizzazione.

Il corretto funzionamento di una rete neurale sull'insieme di apprendimento non offre garanzia di un suo altrettanto corretto funzionamento su dati nuovi, anche se relativi allo stesso concetto ma non utilizzati in fase di apprendimento.

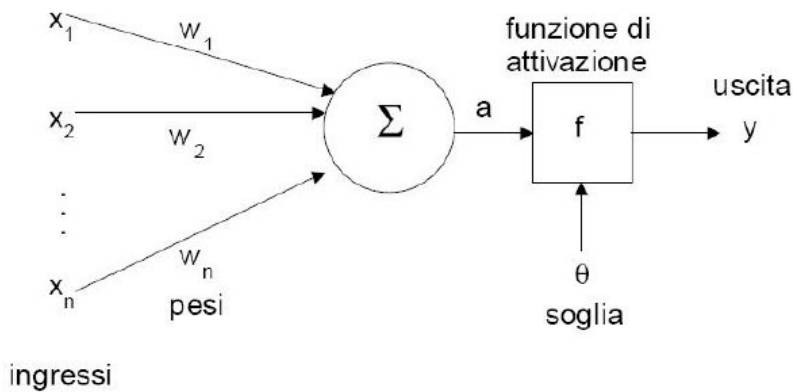
Inoltre occorre sottolineare che l'efficienza di apprendimento dipende anche dall'architettura della rete. In particolare quando il numero delle unità cresce, anche la capacità computazionale cresce mentre la generalizzazione diminuisce: in questo caso non si parla di *fitting* ma di *over-fitting* sull'insieme di apprendimento, ovvero si ha un'enorme quantità di parametri vincolati solo da pochi esempi. Tutto ciò dà luogo ad una sorta di principio di indeterminazione dell'apprendimento: al variare dei pesi della rete non è possibile ottenere un'adeguata generalizzazione per i nuovi esempi. Limitare il numero di ingressi limita dunque l'*over-fitting*.

In realtà la memorizzazione e la capacità di generalizzare possono esser viste in termini di dispendio di risorse di memoria.

L'efficienza di gestione delle risorse disponibili segue il *Principio di Occam*

che può esser enunciato come segue: *date due reti neurali che soddisfino l'insieme di apprendimento, la rete di minor complessità è quella che si comporta meglio su esempi non visti, cioè ha la miglior capacità di generalizzazione.* Tale principio risulta dunque la guida più adatta alla determinazione della struttura della rete neurale ottimale dal punto di vista della generalizzazione.

#### 3.4.4 Il neurone formale e le reti multi-livello



**Figura 3.32:** Neurone artificiale: caratteristiche e costituenti.

Il neurone artificiale detto nodo (vedi Fig. 3.32) è l'unità computazionale atomica di una rete neurale: valuta l'intensità degli ingressi  $x_1, \dots, x_n$  ed a ciascuno assegna un peso  $w_1, \dots, w_n$ , cioè un valore numerico che modula l'impatto (la forza) che tale ingresso ha sulla somma totale, riproducendo il comportamento della sinapsi biologica. Il neurone calcola la somma dei segnali d'ingresso pesati e vi aggiunge la propria *soglia di attivazione*  $\theta$  detta *bias*; successivamente applica la *funzione di trasferimento* o *legge di attivazione*  $f$ , generalmente non lineare, che determina l'uscita  $y$  della rete stessa:

$$y = f(a) = f(\sum_{i=1}^n w_i \cdot x_i - \theta) \quad (3.7)$$

Le funzioni più utilizzate, descritte ampiamente in,<sup>[19]</sup> sono di tipo:

- gradino o *heaviside* con soglia  $\theta$  utilizzata quando la rete deve convertire l'ingresso in un segnale binario

$$\begin{cases} y = 0 & \text{se } a \geq \theta \\ y = 1 & \text{se } a < \theta \end{cases} \quad (3.8)$$

- lineare

$$y = f(a) = a \quad (3.9)$$

- lineare a tratti

- sigmoide con T parametro di temperatura che produce un'uscita compresa tra 0 e 1

$$y = f(a) = \frac{1}{1 + e^{-T \cdot x}} \quad (3.10)$$

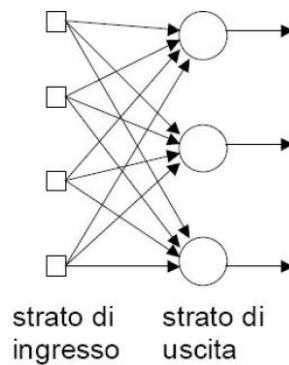
- iperbolica con T parametro di temperatura che produce un'uscita compresa tra 1 e -1

$$y = f(a) = \frac{e^{T \cdot x} - e^{-T \cdot x}}{e^{T \cdot x} + e^{-T \cdot x}} \quad (3.11)$$

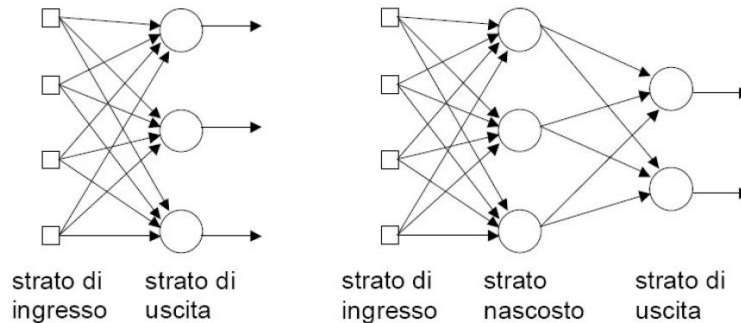
I neuroni vengono organizzati in strati (*layers*): in genere viene utilizzata la stessa funzione di trasferimento per tutti i neuroni che appartengono allo stesso strato della rete ma ciò non è indispensabile. Il comportamento di un neurone è determinato dalla sua funzione di trasferimento e dalle connessioni pesate lungo cui invia e riceve segnali.

Si possono identificare diverse tipologie di architettura di rete: reti completamente connesse dette anche non stratificate, in cui ogni neurone è connesso in modo bidirezionale con tutti gli altri (vedi Fig. 3.33) e reti stratificate (vedi Fig. 3.34). In quest'ultime si individuano degli strati interni tali che ogni neurone appartenente ad essi è connesso con tutti quelli dello strato successivo, ma non esistono connessioni tra i neuroni all'interno dello stesso strato, nè tra neuroni di strati non adiacenti.

Il numero di neuroni e di strati dipendono dal problema specifico. Nello



**Figura 3.33:** Esempio di rete completamente connessa.



**Figura 3.34:** Reti stratificate.

strato d'ingresso non avviene computazione poiché i neuroni d'ingresso devono solo passare i segnali ricevuti dall'esterno allo strato successivo: in questo caso si parla di rete *feedforward*.

Le connessioni tra neuroni di una rete stratificata sono rappresentate mediante tante matrici, contenenti i pesi delle connessioni, quante le coppie di strati adiacenti.

L'architettura di rete maggiormente utilizzata è il *perceptrone* multi-strato (*Multi-Layer Perceptron*, MLP), costituita da uno strato di neuroni d'ingresso, da uno o più strati nascosti e da uno strato di uscita (*output layer*). Bisogna osservare che le reti neurali MLP, approssimanti universali di funzioni non lineari che trasformano lo spazio degli ingressi nello spazio delle uscite, si dividono in reti statiche e reti dinamiche. Si definiscono statiche quelle reti in cui l'uscita ad un determinato istante  $t$  dipende unicamente dall'ingresso allo stesso istante di tempo. Tuttavia in un processo di identificazione è importante saper costruire anche modelli per sistemi dinamici in cui l'uscita in un determinato istante dipende non solo dagli ingressi correnti ma anche dal comportamento precedente del sistema: in sostanza le reti dinamiche sono dotate di memoria.

Per quanto riguarda le modalità di attivazione dei neuroni, a seconda dei modelli di rete, può attivarsi un solo neurone per volta ovvero tutti contemporaneamente: nel primo caso si parla di attivazione asincrona, nel secondo di attivazione sincrona o parallela.

L'addestramento o *training*, procedimento per la stima dei pesi da utilizzare sulle varie connessioni, è un concetto fondamentale della computazione

neurale. Il metodo più utilizzato consiste nel dare in ingresso alla rete un insieme di esempi (*training set*): la risposta fornita dalla rete viene confrontata con la risposta desiderata e dopo aver valutato la differenza (errore) fra le due, proprio in base a tale differenza si aggiustano i pesi. Tale processo è ripetuto in modo iterativo sull'intero *set*, fino a quando le uscite della rete producono un errore inferiore alla soglia stabilita (*goal*). La funzione degli errori fra le risposte della rete e quelle del sistema reale è detta cifra di merito  $C$ . Per un determinato *training dataset*, costituito dalle coppie  $(x_i, y_i)$  si ha:

$$C(E_i) = C(y_i(x_i) - y_i^{NN}(x_i, \Theta)) \quad (3.12)$$

con  $x_i$  vettore degli ingressi,  $y_i$  risposta del sistema reale all'ingresso  $i$ -esimo,  $y_i^{NN}$  risposta della rete all'ingresso  $i$ -esimo e  $\Theta$  vettore dei parametri della rete. Nel presente lavoro è stata utilizzata come cifra di merito la media degli errori quadratici (*Mean Squared Error*, MSE). L'ottimo dei parametri delle rete si ha in corrispondenza del minimo di  $C(\Theta)$ , ricavato in modo iterativo, tale che:

$$\Theta^{i+1} = \Theta^i + \mu^i \cdot f^i \quad (3.13)$$

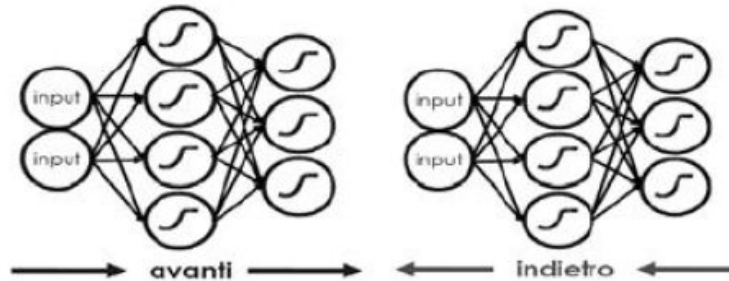
dove  $i$  indica la l'iterazione o epoca,  $f^i$  è il parametro di direzione di ricerca e  $\mu^i$  è il passo o *learning rate* dell'iterazione.

Durante l'allenamento di una rete viene cercato dunque il punto di minimo della superficie dell'errore nello spazio dei pesi. In generale:

$$\begin{cases} f^i = -\nabla(i) \\ \Theta^{i+1} = \Theta^i + \mu^i \cdot Q(-\nabla(i)) \end{cases} \quad (3.14)$$

dove  $Q$  è la matrice che cambia la direzione di ricerca, caratteristica di ogni algoritmo di allenamento.

Ci sono tre algoritmi di addestramento: quelli con supervisione (*supervised learning*), quelli senza supervisione (*unsupervised learning*) e quelli per rinforzo (*reinforcement learning*). Quelli con supervisione modificano i pesi in modo che a definiti valori d'ingresso (*campione d'ingresso*) corrisponda delle uscite definite a priori (*campione d'uscita*). Il più famoso di questi algoritmi è quello di *Back-Propagation*, introdotto nel 1986 e riportato in Fig. 3.35. Tale algoritmo prevede che, per ogni *training set*, i segnali viaggino in una prima fase (a sinistra in figura) dall'ingresso all'uscita, per il calcolo della risposta della rete. Nella seconda fase (a destra in figura) i segnali d'errore vengono propagati all'indietro sulle stesse connessioni, in modo da modificare i pesi, inizializzati in modo casuale.



**Figura 3.35:** Fasi dell'algoritmo di *Back-Propagation*.

Per quanto riguarda invece l'algoritmo di addestramento non supervisionato, i pesi vengono aggiustati in base ad alcune valutazioni sulla qualità del campione d'ingresso da parte delle funzioni dello stesso algoritmo. Dopo questa fase di addestramento è prevista una fase di validazione (valutazione della capacità di generalizzare) ed una di test (valutazione delle prestazioni).

### 3.4.5 Parametri di interesse e scelta delle prese di misura

Nell'ambito di questo lavoro sono stati utilizzati gli algoritmi *Bayesian Regularization* e *Early Stopping* che oltre ad allenare le reti nel *dataset* d'ingresso, migliorano la capacità di generalizzazione minimizzando, proprio durante la fase di *training*, l'errore commesso sull'insieme di dati di validazione detto *validation set*. Con il metodo *Early Stopping* si usano sia insiemi di *training* che di *validation*; con il metodo *Bayesian Regularization* i due insiemi vengono uniti in quanto è sufficiente un unico insieme di dati.

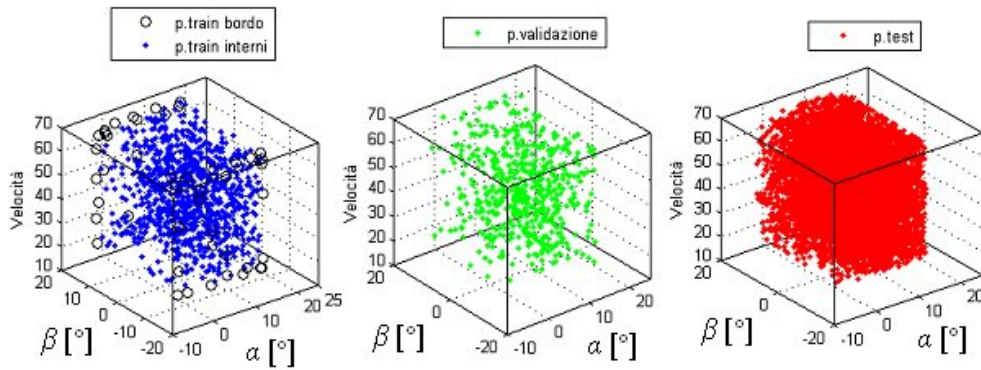
Il *database* di pressioni proveniente dall'analisi fluidodinamica, è stato opportunamente ampliato in modo da migliorare l'accuratezza dell'allenamento, utilizzando *look-up table* tridimensionali ovvero blocchi *Simulink* che interpolano funzioni in tre variabili. Partendo dagli ingressi  $(\alpha, \beta, v)$ , vengono generati dei segnali *random*, opportunamente limitati all'interno dell'involuppo di volo considerato attraverso un blocco di saturazione. Le pressioni nei punti  $(\alpha, \beta, v)$  in cui non è stata effettuata simulazione CFD vengono dunque ottenute tramite interpolazione dalla *look-up table*. Nel lavoro sono state costruite 84 *look-up table* pari al numero di punti di controllo: ad ognuno di essi è associato un vettore che riporta la quaterna  $(\alpha, \beta, v, p)$  in cui  $p$  indica la pressione in uscita dalla *look-up table*. Il *database* così ottenuto è stato suddiviso successivamente in *training set*, *validation* e *testing set*.



Per ottimizzare il *training* occorre, inoltre, scegliere un numero sufficientemente grande di punti ed omogeneamente distribuiti nello spazio: per ottenere ciò si è pensato di suddividere il *database* in tanti elementi cubici di piccole dimensioni detti *bin* (la metodologia è ampiamente descritta in<sup>[15]</sup>). Da studi riportati in letteratura risulta che il rapporto ottimale tra il numero di punti per l'allenamento e quelli per il *test* deve essere circa 1 : 8. Per questo motivo dei 10000 punti totali, 1010 punti costituiscono il *training set*, 540 sono utilizzati per la validazione durante l'allenamento ed infine 8450 per testare le reti già allenate e validate.

Per migliorare ulteriormente l'allenamento, è stato necessario inserire in tale *set* anche i punti di bordo: questi punti sono stati individuati come intersezioni tra un guscio esterno al *set* di allenamento, creato con il comando *convhlln* di *Matlab* ed il *set* stesso. Al contrario il *validation set* pur essendo costruito con lo stesso procedimento del *training set*, non contiene punti di bordo; infine i punti che non appartengono ai precedenti due insiemi, vanno a costituire il *testing set* (vedi Fig. 3.36).

Risulta necessario normalizzare i dati provenienti dall'analisi fluidodinamica



**Figura 3.36:** Visualizzazione dell'insieme di allenamento, validazione e test.

prima di esser utilizzati dalle reti.

La relazione utilizzata è la seguente:

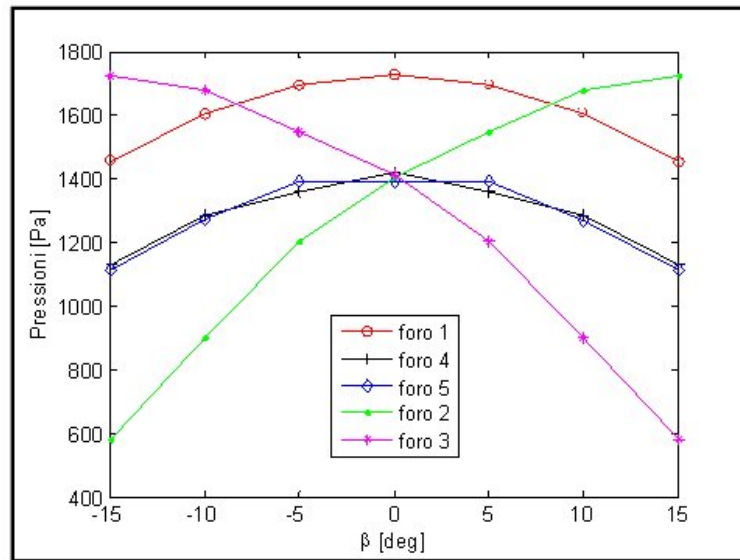
$$v_i^n = \frac{v_i - v_i^{min}}{v_i^{max} - v_i^{min}} \quad (3.15)$$

dove  $v_i^n$  rappresenta il generico normalizzato della pressione, dell'angolo di incidenza o di sideslip,  $v_i^{max}$  e  $v_i^{min}$  i relativi valori massimi e minimi. In particolare per quanto riguarda la normalizzazione della pressione, si è fatto riferimento al sensore centrale indicato dal punto 0, poiché è il sensibile a

tali variazioni. Alla fine di ogni simulazione occorre effettuare il passaggio inverso e riportare le uscite ai valori reali.

Lo studio dell'andamento della pressione, in funzione di  $\alpha$  (vedi Fig. 3.37) e di  $\beta$  (vedi Fig. 3.38), a velocità costante, ha permesso di individuare la posizione ottimale delle prese di misura (fori numerati da 1 a 5), sulla calotta sferica della sonda.

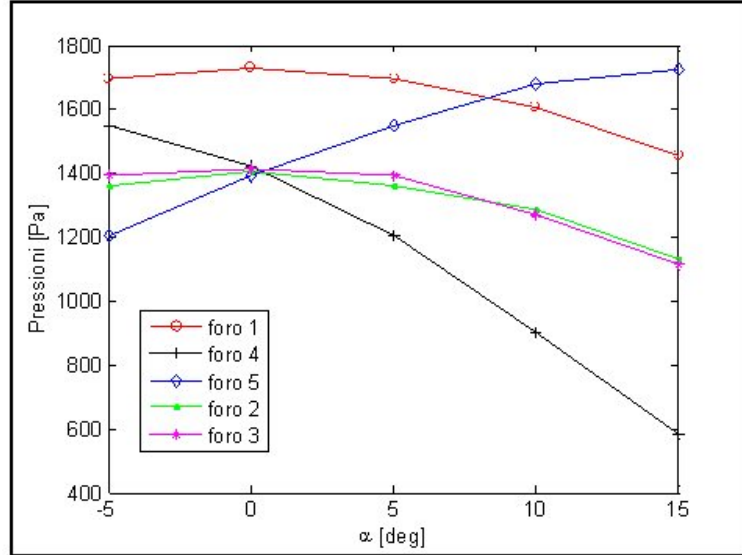
E' stata effettuata un'ampia campagna di prove al fine di individuare la



**Figura 3.37:** Andamento della pressione in funzione di  $\beta$  e delle prese di misura.

struttura neurale capace di ricostruire al meglio i parametri di volo: è stato variato il numero di *layer*, il numero di neuroni, le funzioni di trasferimento, la tipologia ed il numero di ingressi.

Sulla base degli studi presenti in letteratura e delle esperienze maturate in precedenza presso il Dipartimento è stato deciso di utilizzare reti con due strati nascosti contenenti ognuno dieci neuroni. Bisogna sottolineare che, mentre è possibile definire il numero di ingressi ed uscite, non esiste un criterio rigoroso per definire il numero ottimale di strati intermedi e dei neuroni di questi strati. Dunque la scelta è stata operata dopo una serie di prove e sulla base dell'esperienza maturata in applicazioni analoghe. Un numero elevato di strati nascosti comporta un incremento delle dimensioni del training set e del numero di connessioni, con relativo aumento dei tempi di addestramento. La rete impara troppo e non è capace di generalizzare. Al contrario un numero



**Figura 3.38:** Andamento della pressione in funzione di  $\alpha$  e delle prese di misura.

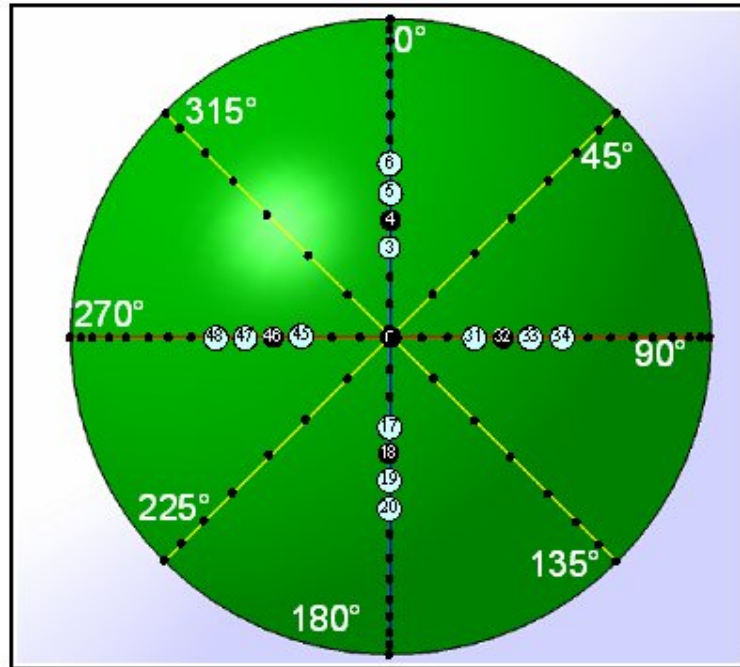
limitato di neuroni e di strati nascosti non permette alla rete di apprendere gli esempi del *training set*.

Per quanto riguarda ad esempio la scelta del numero di ingressi sono stati scelti i sensori più sensibili alle variazioni di pressione. A tal proposito sono state effettuate delle simulazioni per studiare la sensibilità dei punti riportati in Fig. 3.39 alla  $\Delta P$ : dall'analisi è emerso che i punti più sensibili sono contrassegnati dai numeri 0, 4, 18, 32 e 46. In particolare la Fig. 3.40 mostra l'andamento della pressione al variare di  $\alpha$  a  $\beta = 0^\circ$  e  $v = 50m/s$ , mentre la Fig. 3.41 mostra l'andamento della pressione al variare di  $\beta$  ad  $\alpha = 0^\circ$  e  $v = 50m/s$  nei punti sopracitati.

Le reti utilizzate nel presente lavoro presentano dunque un solo neurone d'uscita e cinque ingressi: la notazione *Simple Input*, SI indica che gli ingressi provengono direttamente dal *database* mentre la notazione *Normalized Simple Input*, NSI indica ingressi normalizzati, ovvero riportati all'intervallo  $[0, 1]$  tramite la Eq. (3.15). La Fig. 3.42 chiarisce tutte le caratteristiche delle reti esaminate.

Di seguito viene riportato il valore di alcuni parametri d'interesse, mantenuti costanti per tutte le reti:

- numero di epoche: 3000



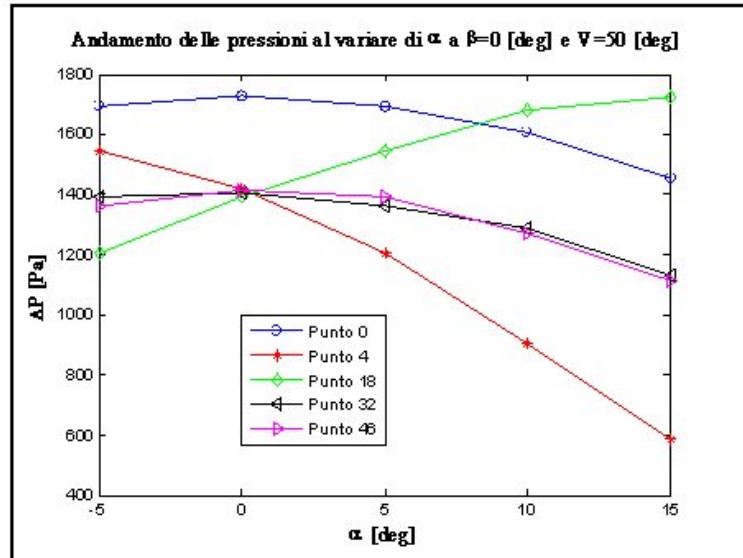
**Figura 3.39:** Punti scelti per posizionare i sensori (ingressi ottimali per le reti neurali).

- valore della cifra di merito (goal):  $10^{-5}$
- parametro di direzione  $\mu$ :  $10^{-3}$

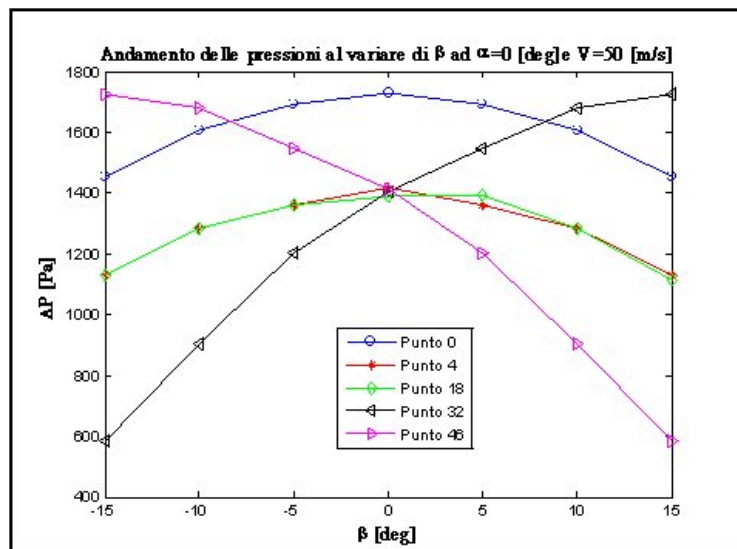
Nei paragrafi che seguono sono riportati gli errori commessi dalle reti neurali sia in fase di allenamento che di *test*.

Per quanto riguarda la fase di test, necessaria per valutare la capacità delle reti di generalizzare, si è scelto di riportare sia l'errore massimo che l'errore medio.

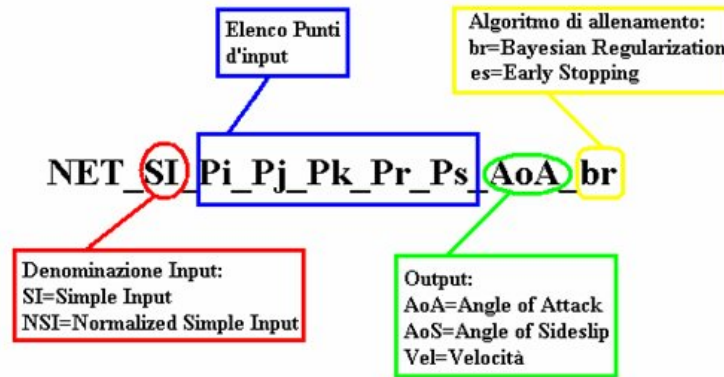
Per la fase di allenamento, se questa è stata eseguita con il metodo *Bayesian Regularization* l'errore viene calcolato come somma dei quadrati degli errori commessi in ogni punto d'allenamento (*Sum of Squared Error*, SSE); se le reti sono state allenate con il metodo *Early Stopping* il parametro d'errore è il *Mean of squared Error*, MSE, calcolato come media degli errori commessi nei punti d'allenamento, elevati al quadrato. Inoltre si è ritenuto opportuno riportare l'errore massimo e l'errore medio commesso sul *training set*, la distribuzione degli errori in funzione di  $\alpha$ ,  $\beta$ ,  $v$  ed il numero di eccedenze (ovvero il numero di volte che l'errore supera  $0.5^\circ$  e  $0.1\text{m/s}$  per la velocità,



**Figura 3.40:** Andamento della pressione  $\Delta P$  in funzione di  $\alpha$  nei punti scelti come sensori di misura.



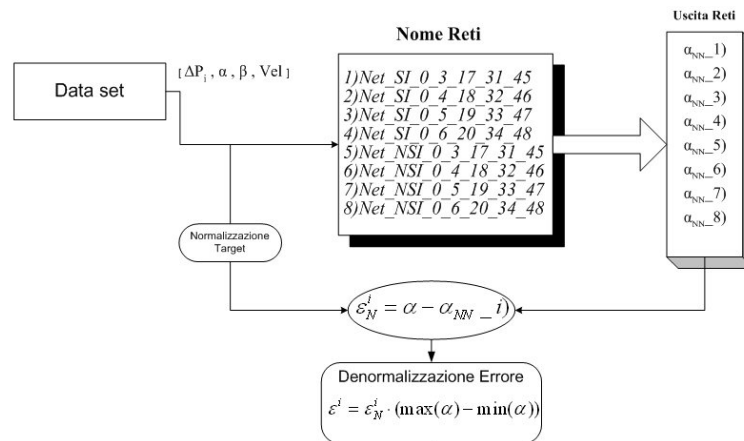
**Figura 3.41:** Andamento della pressione  $\Delta P$  in funzione di  $\beta$  nei punti scelti come sensori di misura.



**Figura 3.42:** Esempio di denominazione di una generica rete.

come risulta da precedenti studi riportati in letteratura).

Il calcolo dell'errore commesso dalle reti nella ricostruzione dell'uscita segue lo schema riportato in Fig. 3.43.



**Figura 3.43:** Schema di calcolo dell'errore commesso dalle reti nella ricostruzione dei parametri di volo.

### 3.4.6 Valutazione degli errori commessi dalle reti a singola uscita

Come accennato in precedenza, questa tipologia di reti ha l'obiettivo di ricostruire l'angolo di incidenza  $\alpha$ , o l'angolo di derapata  $\beta$  oppure la velocità  $v$ . Di seguito vengono elencate alcune caratteristiche e gli errori ottenuti:

- *training set*: 1010 punti;
- *validation set*: 540 punti;
- *testing set*: 8450 punti;
- dominio dell'uscita  $\alpha$ :  $[-5^\circ, 15^\circ]$ ;
- dominio dell'uscita  $\beta$ :  $[-15^\circ, 15^\circ]$ ;
- dominio dell'uscita  $v$ :  $[20m/s, 70m/s]$ ;

Le figure riportano l'uscita della rete, il tipo ed il numero di ingressi, la due metodologie di allenamento utilizzate ed infine gli errori commessi sul *training set* e sul *testing set*.

Alcune osservazioni sulle reti a singola uscita:

#### **Reti che ricostruiscono $\alpha$**

- l'errore SSE/MSE si riduce utilizzando il metodo allenamento *Early Stopping*;
- l'errore medio e l'errore massimo si riducono nel caso di reti allenate con il metodo *Bayesian Regularization*, BR;
- la distribuzione dei picchi è relativa alla rete ha presentato il più basso errore medio sul *testing set*. Tali picchi si presentano in generale a basse velocità e ai bordi dei domini, indipendentemente dall'uscita esaminata;
- il numero delle eccedenze delle reti allenate con il metodo *Early Stopping*, ES è di un ordine di grandezza superiore a quelle delle reti allenate con il *Bayesian Regularization*: ciò spiega l'errore medio più alto delle reti ES.

#### **Reti che ricostruiscono $\beta$**

- l'errore SSE/MSE si riduce rispetto alle reti su  $\alpha$  se si utilizza il metodo allenamento BR;
- le reti BR ed ES presentano errori massimi e medi più bassi sul *training set*, rispetto al *testing set*;
- come per le reti su  $\alpha$  il numero di eccedenze delle reti ES è maggiore rispetto a quello delle reti BR.

### Reti che ricostruiscono $v$

- anche per questa tipologia di rete si possono ripetere i commenti fatti per le precedenti;
- l'errore massimo delle reti BR sul *testing set* è abbastanza elevato: questo può esser legato ad eventuali problemi di inizializzazione delle simulazioni. Ciò può esser risolto ripetendo più volte l'allenamento o infittendo l'insieme di dati. A tal proposito si osservino i nuovi risultati di Fig. 3.43.
- si osserva che le eccedenze delle reti BR ed ES sono simili e l'errore percentuale massimo è più basso rispetto a quello delle reti su  $\alpha$  o su  $\beta$ . Ciò deriva dal fatto che tale errore viene calcolato su tutto il dominio della variabile d'uscita e nel caso della velocità questo dominio è piuttosto ampio.

#### 3.4.7 Valutazione degli errori commessi dalle reti a doppia uscita

In questo paragrafo vengono riportati i risultati ottenuti nella ricostruzione contemporanea delle uscite  $\alpha$  e  $\beta$ : la struttura generale della rete resta comunque analoga a quella delle reti a singola uscita.

I parametri presi in esame sono gli stessi dell'analisi precedente (ingressi, caratteristiche dei *database*, SSE/MSE, errore massimo e medio, eccedenze, ...).

### Reti che ricostruiscono $\alpha$ e $\beta$

- l'errore SSE/MSE delle reti BR è dell'ordine di  $10^{-1}$  mentre quello delle reti allenate con il metodo ES è dell'ordine di  $10^{-4}$ ;
- l'errore medio dell'ordine di  $10^{-1}$  per entrambe le uscite e gli allenamenti;
- l'errore massimo più elevato se si utilizza l'allenamento BR;
- come nelle analisi precedenti, la distribuzione dei picchi è relativa alla rete ha presentato il più basso errore medio sul *testing set*. Tali picchi si presentano ai bordi dei domini e alle basse velocità.

## 3.5 Risultati generali

Dalle simulazioni effettuate sulle diverse tipologie di reti neurali utilizzate per la ricostruzione dei parametri  $\alpha$ ,  $\beta$  e  $v$  emerge che:



- La scelta di posizionare le prese di misura sulla calotta sferica della sonda è risultata ottimale. Infatti le reti neurali, allenate sui valori di pressione locale forniti da tali prese, consentono una ricostruzione soddisfacente di  $\alpha$ ,  $\beta$  e  $v$ . In futuro si pensa di estendere gli studi effettuati, per la ricostruzione della pressione stessa, momentaneamente affidata ad un tubo di *Pitot*;
- Non si notano differenze sostanziali nei risultati ottenuti dalla reti che usano in ingresso i valori forniti alle prese (0, 5, 19, 33, 47) e quelle che usano le prese (0, 6, 20, 34, 48);
- La normalizzazione degli input non cambia l'ordine di grandezze dei risultati;
- L'allenamento mediante il metodo *Early Stopping* è molto più veloce di quello effettuato mediante il *Bayesian Regularization*: il numero di epoche e lo stesso indice MSE è notevolmente più basso del corrispettivo SSE;
- Le reti allenate con il BR, rispetto a quelle che usano il metodo ES, riescono a generalizzare meglio su punti diversi da quelli di allenamento;
- Le reti a singola uscita commettono errori medi e massimi sul training set più bassi di quelli commessi dalle reti a doppia uscita. Ciò non vale sul *testing set* dove si comportano meglio le reti a doppia uscita.

E' possibile giustificare tale comportamento facendo una piccola precisazione. Il numero di parametri di una rete è uguale alla somma del numero di pesi e delle soglie (*bias*) d'attivazione dei neuroni. Il numero di pesi dipende dal numero di neuroni in uscita, il numero di *bias* è uguale al numero di neuroni. Nel caso di reti a singola uscita, il numero di parametri è 181 mentre nel caso di reti a due uscite è 192: dunque realizzare due reti a singola uscita comporta un numero di parametri doppio rispetto ad un'unica rete a doppia uscita.

Le reti a doppia uscita comportano quindi un notevole risparmio di memoria: in presenza di eventuali avarie si perdono, però, entrambe i segnali.

Le figure che seguono riassumono il numero di epoche necessarie per l'allenamento con il metodo BR e il metodo ES, relativi alla rete sulla velocità, con ingressi normalizzati, provenienti dalle prese (0 5 19 33 47).

Si osservi che il numero di parametri effettivo (168 in figura) è minore di quello teorico detto in precedenza, poiché la rete cerca sempre di ridurre al

minimo la matrice dei pesi, eliminando di volta in volta i valori che non ritiene necessari.

### 3.6 Aspetti innovativi dell'argomento di ricerca trattato

Come detto nei paragrafi precedenti, la sonda sviluppata nel corso del presente dottorato di ricerca si ispira ai sistemi FADS, ovvero sistemi di sensori dati aerea di tipo non intrusivo, dotati di un numero elevato di prese di pressione, realizzate direttamente sulla parte prodiera della fusoliera.

Il numero elevato di prese di pressione che caratterizza tali sistemi ne consente il funzionamento anche in caso di perdita di una o più misure dovute al verificarsi di avarie. Quest'aspetto rende i FADS particolarmente adatti a sistemi di tipo FCS-FBW, presentati brevemente nel primo capitolo e caratterizzati da architetture a molteplice ridondanza.

Per quanto esposto in precedenza, i FADS trovano applicazione su velivoli da rientro spaziale di nuova generazione come lo spaziosplano X-33, su velivoli di tipo *Stealth* e su velivoli non pilotati.

Infatti le particolari condizioni in cui si trovano ad operare i velivoli da rientro spaziale rendono praticamente impossibile l'impiego di sensori di tipo intrusivo che sarebbe soggetti ad eccessivo riscaldamento in fase di rientro in atmosfera. Per quanto riguarda invece i velivoli *Stealth*, l'invisibilità alle difese nemiche non può essere limitata alla sola segnatura radar, ma si estende anche alla segnatura termica (principalmente data dagli scarichi dei motori) ed alle emissioni elettroniche (provenienti dal radar e dagli strumenti di bordo ecc.), nonché alla visibilità ottica del velivolo stesso. Recentemente sono state messe a punto delle tecniche che permettono di individuare la turbolenza prodotta da un generico velivolo: a tal proposito risulta importante garantire una *low observability* eliminando qualunque superficie capace di modificare il flusso intorno al velivolo e creare turbolenza aggiuntiva.

Inoltre nel caso di velivoli non pilotati i sistemi di bordo devono garantire bassi costi ed ingombri limitati.

Occorre sottolineare che la stima dei parametri di volo a partire da un numero elevato di segnali disponibili rappresenta un problema piuttosto complesso e comporta la necessità di monitorare tali segnali per individuare eventuali avarie.

Negli ultimi tempi si è sviluppato un notevole interesse, anche dal punto di vista industriale, nei confronti di tale tipologia di sensori: ne è testimonianza ad esempio il progetto europeo nEUROn, per lo sviluppo di un velivolo da combattimento non pilotato dalle accentuate caratteristiche *stealth*.

Nonostante questo interesse la letteratura risulta ancora povera di lavori sull'argomento. Il presente dottorato getta dunque le basi per una ricerca in tal senso. Dopo aver ipotizzato una possibile geometria del sensore-test, questa è stata analizzata tramite studi di fluidodinamica computazionale per ottimizzarla ed ottenere un *database* campione, che lega le pressioni locali con le condizioni di volo.

Tale *database* è stato utilizzato per allenare le reti neurali, che costituiscono un approccio promettente per la ricostruzione dei parametri di volo.

Le reti neurali richiedono infatti un numero di parametri da memorizzare inferiore rispetto ad un approccio di tipo classico ed inoltre risultano facilmente riconfigurabili su nuovi dati di ingresso.

A differenza di intelligenze artificiali classiche, come i sistemi esperti, hanno la capacità di far tesoro delle esperienze precedenti e dei problemi già risolti, auto-modificando il loro comportamento. Questa grande capacità di riconfigurarsi rende più immediata una futura espansione dell'involuppo di volo, senza sconvolgere l'architettura attuale.

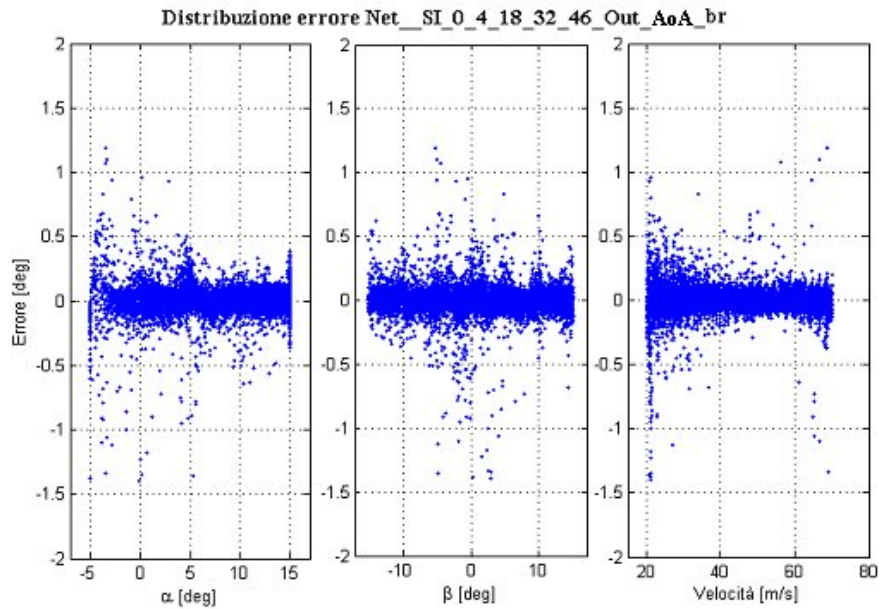
Il lavoro ha quindi valutato la fattibilità di tale approccio limitatamente al calcolo di  $\alpha$ ,  $\beta$ ,  $v$ , impiegando per il momento reti a cinque ingressi.

Gli errori commessi nella ricostruzione dei parametri di volo sia sull'insieme di addestramento che di test sono soddisfacenti: ciò conferma la bontà dell'architettura sviluppata e della metodologia utilizzata per la scelta di tali insiemi.

Il lavoro svolto può esser considerato dunque come la base di partenza per lo sviluppo di un'architettura neurale più complessa, con un numero maggiore di ingressi, capace di individuare le avarie e gestire le ridondanze del sistema.

Output	Input	Metodo di allenamento	Training set			Testing set	
			SSE/MSE	Errore Massimo	Errore Medio	Errore Massimo	Errore Medio
$\alpha$	Net_SI_Pi_Pj_Pk_Pr_Ps_AoA_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,017306	0,447420	0,061806	4,708000	0,104670
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,009211	0,309000	0,044788	4,314300	0,069848
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,018390	0,652410	0,058578	11,619000	0,093096
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,019264	0,557770	0,064088	2,997800	0,087830
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoA_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,011873	0,427340	0,047425	2,701900	0,072433
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,022973	0,653120	0,063271	11,798000	0,096694
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,015694	0,581250	0,053015	3,570400	0,088107
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,012754	0,437120	0,049785	3,178600	0,092618
	Net_SI_Pi_Pj_Pk_Pr_Ps_AoA_es						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000941	3,749000	0,471450	17,255000	0,752190
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,00050071	3,2339	0,31845	7,9663	0,47914
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000399	2,721100	0,298050	6,499800	0,447190
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,006979	6,958300	1,269900	7,660600	1,393700
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoA_es						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000062	0,958130	0,110110	3,568700	0,152030
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000202	1,932400	0,175050	4,524000	0,207690
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,00019901	2,5759	0,18295	4,6833	0,24017
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000377	3,190300	0,267930	4,749000	0,326950

**Figura 3.44:** Tabella degli errori (in  $[\circ]$ ) commessi dalle reti nella ricostruzione di  $\alpha$ .



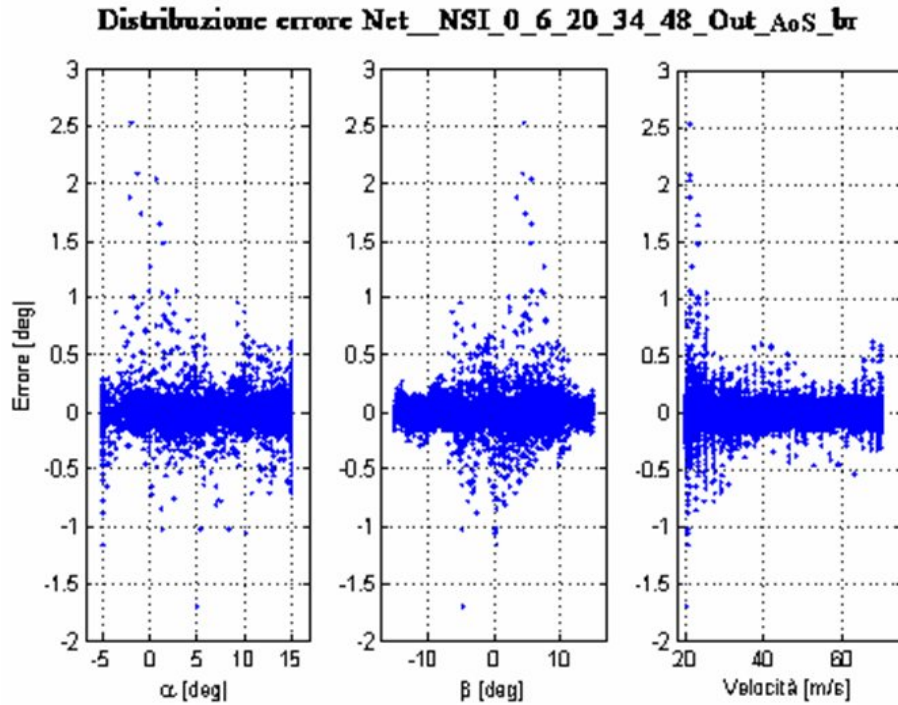
**Figura 3.45:** Distribuzione, nell'involuppo  $(\alpha, \beta, v)$ , degli errori (in  $^{\circ}$ ) commessi dalla rete su  $\alpha$ .

Nome Rete	Metodo di allenamento	Eccedenze	Errore Max Percentuale (%)
1)Net_NSI_0_3_17_31_45_Out_AoA_br	Bayesian Regularization	83	13,51
2)Net_NSI_0_4_18_32_46_Out_AoA_br		146	58,99
3)Net_NSI_0_5_19_33_47_Out_AoA_br		148	17,85
4)Net_NSI_0_6_20_34_48_Out_AoA_br		275	15,89
5)Net_SI_0_3_17_31_45_Out_AoA_br		157	23,54
6)Net_SI_0_4_18_32_46_Out_AoA_br		80	21,57
7)Net_SI_0_5_19_33_47_Out_AoA_br		134	58,10
8)Net_SI_0_6_20_34_48_Out_AoA_br		100	14,99
1)Net_NSI_0_3_17_31_45_Out_AoA_es	Early Stopping	434	17,84
2)Net_NSI_0_4_18_32_46_Out_AoA_es		765	22,62
3)Net_NSI_0_5_19_33_47_Out_AoA_es		2603	23,42
4)Net_NSI_0_6_20_34_48_Out_AoA_es		1682	23,75
5)Net_SI_0_3_17_31_45_Out_AoA_es		3925	86,28
6)Net_SI_0_4_18_32_46_Out_AoA_es		7672	39,83
7)Net_SI_0_5_19_33_47_Out_AoA_es		2670	32,50
8)Net_SI_0_6_20_34_48_Out_AoA_es		6697	38,30

**Figura 3.46:** Tabella delle eccedenze delle reti che ricostruiscono  $\alpha$ .

Output	Input	Metodo di allenamento	Training set			Testing set	
			SSE/MSE	Errore Massimo	Errore Medio	Errore Massimo	Errore Medio
$\beta$	Net_SI_Pi_Pj_Pk_Pr_Ps_AoS_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,011152	0,764020	0,069483	4,455600	0,105650
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,011031	0,594620	0,070484	9,266900	0,118600
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,008321	0,598160	0,063145	4,057400	0,100570
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,020640	0,760260	0,103840	23,727000	0,179250
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoS_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,011685	0,776740	0,069462	5,347000	0,097457
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,009937	0,574170	0,066164	7,574800	0,102130
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,006967	0,609290	0,055964	10,033000	0,094088
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,006619	0,379380	0,055509	2,526300	0,082201
	Net_SI_Pi_Pj_Pk_Pr_Ps_AoS_es						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000875	5,198600	0,924730	11,976000	1,002800
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000444	3,331600	0,470510	15,160000	0,715280
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000116	3,178600	0,212100	5,204100	0,229100
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000743	3,096400	0,608880	4,601700	0,620590
	Net_NSI_Pi_Pj_Pk_Pr_Ps_AoS_es						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000210	3,825400	0,293820	25,922000	0,373280
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000300	3,817700	0,385740	5,116200	0,417430
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000165	3,243200	0,259680	5,259700	0,301640
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000122	3,054500	0,218050	5,817700	0,237200

**Figura 3.47:** Tabella degli errori (in  $^{\circ}$ ) commessi dalle reti nella ricostruzione di  $\beta$ .



**Figura 3.48:** Distribuzione, nell'involuppo  $(\alpha, \beta, v)$ , degli errori (in  $^{\circ}$ ) commessi dalle reti nella ricostruzione di  $\beta$ .

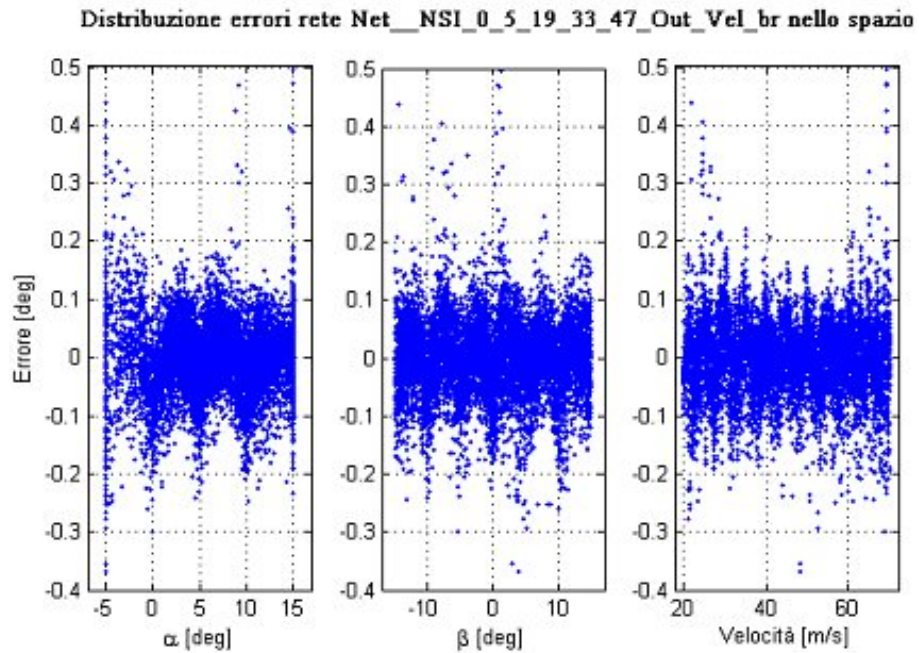
Nome Rete	Metodo di allenamento	Eccedenze	Errore Max Percentuale (%)
1)Net_NSI_0_3_17_31_45_Out_AoA_br	<i>Bayesian Regularization</i>	83	13,51
2)Net_NSI_0_4_18_32_46_Out_AoA_br		146	58,99
3)Net_NSI_0_5_19_33_47_Out_AoA_br		148	17,85
4)Net_NSI_0_6_20_34_48_Out_AoA_br		275	15,89
5)Net_SI_0_3_17_31_45_Out_AoA_br		157	23,54
6)Net_SI_0_4_18_32_46_Out_AoA_br		80	21,57
7)Net_SI_0_5_19_33_47_Out_AoA_br		134	58,10
8)Net_SI_0_6_20_34_48_Out_AoA_br		100	14,99
1)Net_NSI_0_3_17_31_45_Out_AoA_es	<i>Early Stopping</i>	1860	86,41
2)Net_NSI_0_4_18_32_46_Out_AoA_es		2470	17,05
3)Net_NSI_0_5_19_33_47_Out_AoA_es		1404	17,53
4)Net_NSI_0_6_20_34_48_Out_AoA_es		965	19,39
5)Net_SI_0_3_17_31_45_Out_AoA_es		5794	39,92
6)Net_SI_0_4_18_32_46_Out_AoA_es		3690	50,53
7)Net_SI_0_5_19_33_47_Out_AoA_es		881	17,35
8)Net_SI_0_6_20_34_48_Out_AoA_es		4091	15,34

**Figura 3.49:** Tabella delle eccedenze delle reti che ricostruiscono  $\beta$ .

Output	Input	Metodo di allenamento	Training set			Testing set	
			SSE/MSE	Errore Massimo	Errore Medio	Errore Massimo	Errore Medio
Vel	Net_SI_Pi_Pj_Pk_Pr_Ps_Vel_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,008700	0,874600	0,110300	12,391000	0,140410
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	85,050300	24,710000	12,516000	24,781000	12,277000
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,045200	1,057700	0,242900	29,735600	0,508900
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	86,266000	25,086600	12,600200	25,080000	12,353000
	Net_NSI_Pi_Pj_Pk_Pr_Ps_Vel_br						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,008693	0,141600	0,035600	0,854880	0,050528
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,008850	0,158100	0,037000	0,547790	0,045950
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,009110	0,159400	0,037700	0,497110	0,049604
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,007840	0,133400	0,034400	0,725550	0,046050
	Net_SI_Pi_Pj_Pk_Pr_Ps_Vel_es						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000068	2,166500	0,309160	4,631300	0,355100
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000010	0,883550	0,115870	4,066600	0,177210
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000001	0,92695	0,13192	1,7924	0,15378
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000249	3,539800	0,640040	9,267400	0,754450
	Net_NSI_Pi_Pj_Pk_Pr_Ps_Vel_es						
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm	0,000031	1,538100	0,208610	2,493600	0,240780
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	0,000010	0,734260	0,122780	4,325600	0,168680
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm	0,000010	1,151000	0,116330	2,091700	0,157390
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm	0,000024	1,742800	0,216580	3,134000	0,267240

**Figura 3.50:** Tabella degli errori (in [m/s]) commessi dalle reti nella ricostruzione di  $v$ .





**Figura 3.51:** Distribuzione, nell'involuppo  $(\alpha, \beta, v)$ , degli errori (in  $^{\circ}$ ) commessi dalle reti nella ricostruzione di  $v$ .

Nome Rete	Metodo di allenamento	Eccedenze	Errore Max Percentuale (%)
1)Net_NSI_0_3_17_31_45_Out_Vel_br	Bayesian Regularization	987	1,71
2)Net_NSI_0_4_18_32_46_Out_Vel_br		752	1,10
3)Net_NSI_0_5_19_33_47_Out_Vel_br		903	0,99
4)Net_NSI_0_6_20_34_48_Out_Vel_br		718	1,45
5)Net_SI_0_3_17_31_45_Out_Vel_br		4190	24,78
6)Net_SI_0_4_18_32_46_Out_Vel_br		8451	49,56
7)Net_SI_0_5_19_33_47_Out_Vel_br		6306	59,47
8)Net_SI_0_6_20_34_48_Out_Vel_br		8422	50,16
1)Net_NSI_0_3_17_31_45_Out_Vel_es	Early Stopping	5836	4,99
2)Net_NSI_0_4_18_32_46_Out_Vel_es		4623	8,65
3)Net_NSI_0_5_19_33_47_Out_Vel_es		4563	4,18
4)Net_NSI_0_6_20_34_48_Out_Vel_es		6252	6,27
5)Net_SI_0_3_17_31_45_Out_Vel_es		6675	9,26
6)Net_SI_0_4_18_32_46_Out_Vel_es		4803	8,13
7)Net_SI_0_5_19_33_47_Out_Vel_es		4582	3,58
8)Net_SI_0_6_20_34_48_Out_Vel_es		7602	18,53

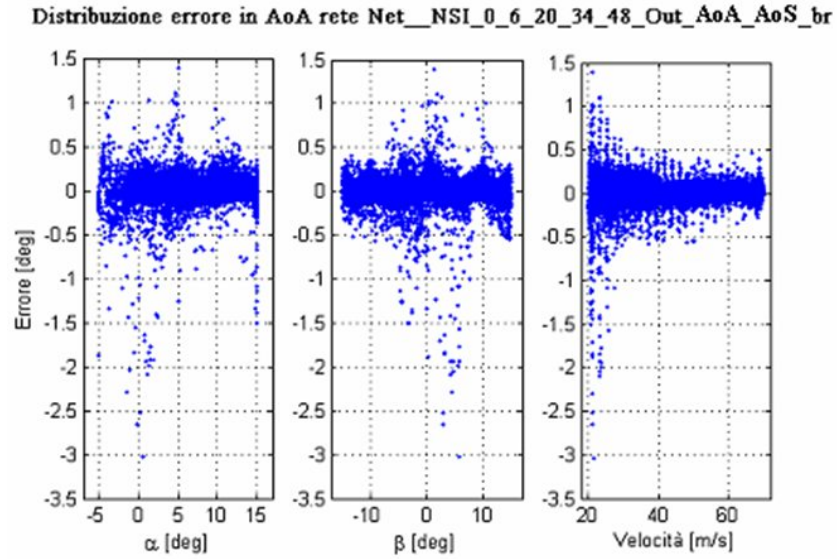
**Figura 3.52:** Tabella delle eccedenze delle reti che ricostruiscono  $v$ .

Output	Input	Metodo di allenamento	Training set			Testing set		
			SSE/MSE	Errore Massimo	Errore Medio	Errore Massimo	Errore Medio	
Vel	Net SI Pi Pj Pk Pr Ps Vel br							
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,0015772	0,15862	0,03993	1,861000	0,048795	
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,001515	0,174390	0,039138	1,905900	0,044708	
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,002999	0,244490	0,055122	0,478380	0,059119	
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,0014504	0,20947	0,037899	0,403620	0,043336	
	Net NSI Pi Pj Pk Pr Ps Vel br							
	$P_0, P_3, P_{17}, P_{31}, P_{45}$	br	0,001492	0,168170	0,038934	0,598960	0,043529	
	$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	0,001412	0,188160	0,037157	0,503690	0,043686	
	$P_0, P_5, P_{19}, P_{33}, P_{47}$	br	0,001601	0,201770	0,040142	0,801010	0,045888	
	$P_0, P_6, P_{20}, P_{34}, P_{48}$	br	0,001099	0,156190	0,032667	0,665400	0,037550	

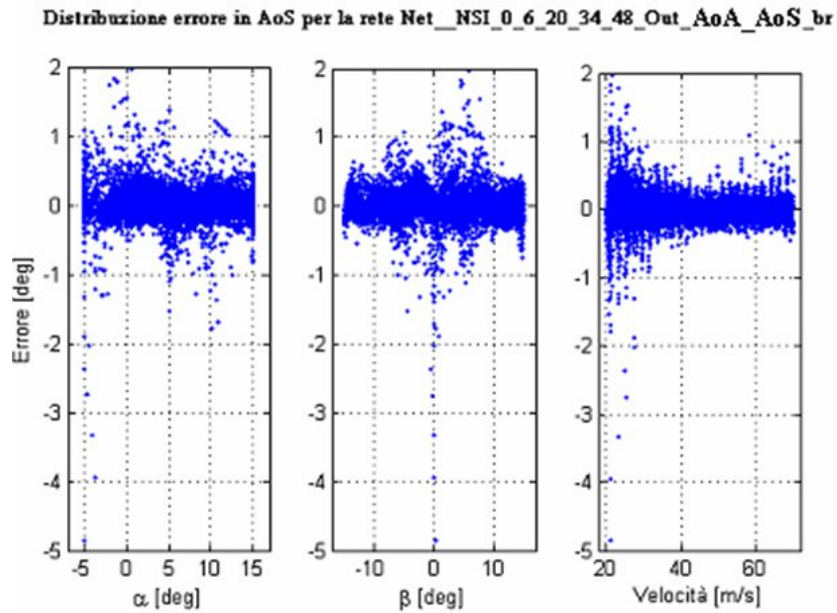
**Figura 3.53:** Tabella degli errori (in [m/s]) commessi dalle reti nella ricostruzione di  $v$  sul *training set* ampliato.

Input	Metodo di training	Output	Training set			Output	Testing set			Training set			Testing set		
			SSE/MSE	Errore Massimo	Errore Medio		Errore Massimo	Errore Medio	Errore Massimo	Errore Medio	Errore Massimo	Errore Medio			
Net_SI_Pi_Pj_Pk_Pr_Ps_AoS_br															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	br		0,0945	0,8825	0,0902		3,5604	0,1229		1,6179	0,1461		4,2448	0,1607	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	$\alpha$	0,1246	0,6762	0,1054		3,5968	0,1233		1,4743	0,1844		4,8618	0,2140	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	br		0,1342	0,9298	0,1188		4,7172	0,1627	$\beta$	1,7574	0,1636		4,4720	0,1927	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	br		0,0990	1,1211	0,0987		3,474	0,13		1,2961	0,1427		4,8859	0,1743	
Net_NSI_Pi_Pj_Pk_Pr_Ps_AoS_br															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	br		0,0834	0,8729	0,0898		2,6855	0,1050		1,4586	0,1293		5,2067	0,1585	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	br	$\alpha$	0,1209	1,0149	0,1072		3,4694	0,1272		1,9711	0,1562		4,1049	0,1863	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	br		0,1168	1,1133	0,1033		4,0192	0,1254	$\beta$	1,5897	0,1465		3,0730	0,1582	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	br		0,0836	0,7603	0,0811		3,0312	0,1048		2,1881	0,1306		4,8454	0,1421	
Net_SI_Pi_Pj_Pk_Pr_Ps_AoS_es															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm		0,0007	4,3831	0,3757		9,8864	0,4178		10,4890	0,4219		13,9550	0,4921	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	$\alpha$	0,0002	3,5472	0,2021		4,5654	0,2346		2,7069	0,2455		6,8439	0,2867	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm		0,0003	3,7392	0,2236		12,4690	0,2662	$\beta$	3,3663	0,3181		9,2975	0,4286	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm		0,0001	1,5817	0,1343		4,0469	0,1736		2,8616	0,2066		8,4399	0,2713	
Net_NSI_Pi_Pj_Pk_Pr_Ps_AoS_es															
$P_0, P_3, P_{17}, P_{31}, P_{45}$	es-lm		0,0003	3,6668	0,2013		4,1985	0,2090		3,5316	0,2767		4,7644	0,2937	
$P_0, P_4, P_{18}, P_{32}, P_{46}$	es-lm	$\alpha$	0,0002	3,0267	0,2000		4,4808	0,2278		3,0795	0,2337		5,1495	0,2569	
$P_0, P_5, P_{19}, P_{33}, P_{47}$	es-lm		0,0004	3,7781	0,2819		4,6664	0,3165	$\beta$	3,3266	0,3302		5,6393	0,3556	
$P_0, P_6, P_{20}, P_{34}, P_{48}$	es-lm		0,0003	3,2081	0,2095		5,2580	0,2599		3,4358	0,3619		18,8960	0,4724	

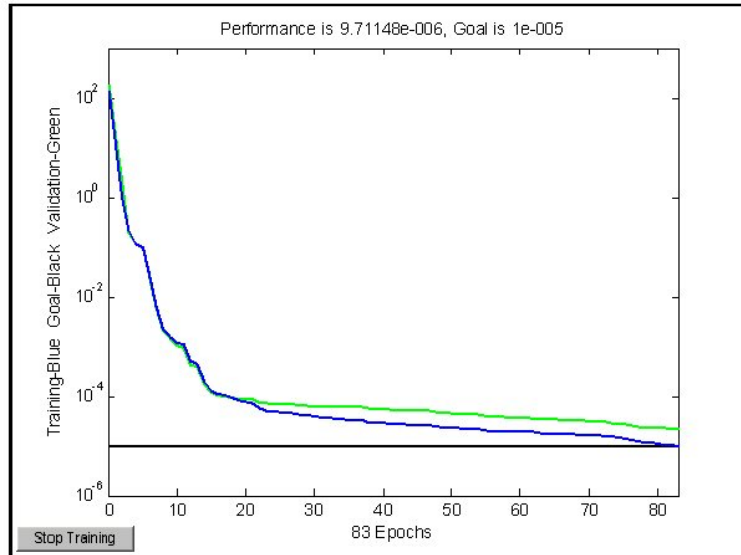
**Figura 3.54:** Tabella degli errori (in [m/s]) commessi dalle reti nella ricostruzione di  $\alpha$  e  $\beta$ .



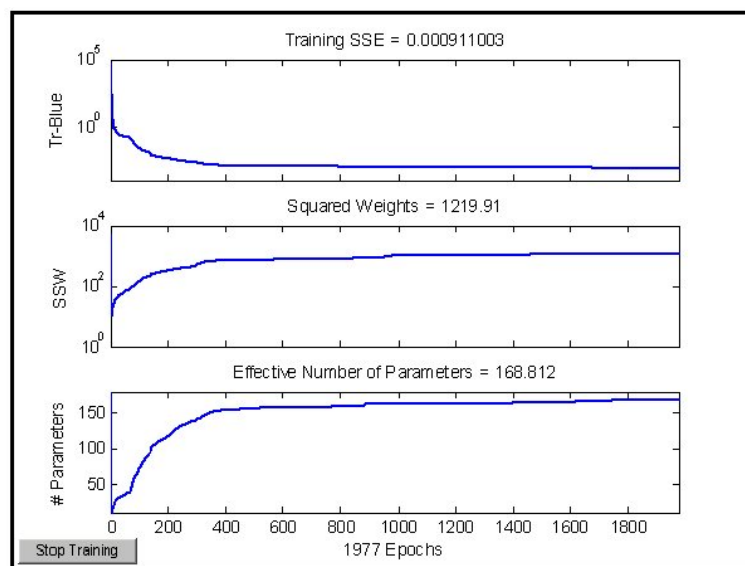
**Figura 3.55:** Distribuzione, nell'involuppo  $(\alpha, \beta, v)$ , degli errori su  $\alpha$  (in  $^{\circ}$ ), commessi dalle reti che ricostruiscono  $\alpha$  e  $\beta$ .



**Figura 3.56:** Distribuzione, nell'involuppo  $(\alpha, \beta, v)$ , degli errori su  $\beta$  (in  $^{\circ}$ ), commessi dalle reti che ricostruiscono  $\alpha$  e  $\beta$ .



**Figura 3.57:** Andamento dell'errore MSE durante l'allenamento della rete sulla velocità, con ingressi normalizzati, provenienti dalle prese (0 5 19 33 47).



**Figura 3.58:** Andamento dell'errore SSE durante l'allenamento della rete sulla velocità, con ingressi normalizzati, provenienti dalle prese (0 5 19 33 47).

## Conclusioni e sviluppi futuri

Nella presente tesi di dottorato è stato affrontato il problema dell'integrazione del sistema dati aria nei moderni sistemi di controllo con trasmissione dei comandi di volo di tipo *Fly-by-Wire*, (FBW).

Nella prima parte del lavoro, è stato presentato un quadro riassuntivo dell'architettura di un sistema *Fly-by-Wire*, definendo obiettivi, componenti, principali applicazioni, problematiche connesse con l'affidabilità e vantaggi associati all'impiego di tale sistema. Successivamente è stato messo in evidenza il ruolo fondamentale svolto dal sistema dati aria e dai parametri di volo nei *computer* di bordo, per assolvere ad esempio a funzioni di protezione dell'involuppo di volo.

Il lavoro descrive le logiche originali sviluppate per l'individuazione delle avarie dei sensori, la gestione delle ridondanze e la riconfigurazione del sistema dati aria. In particolare il lavoro svolto fa riferimento al sistema installato sull'addestratore militare di nuova generazione M346, prodotto dalla società Alenia Aermacchi s.p.a. ed Alenia SIA costituito da quattro sonde multifunzione, auto-allineanti alla direzione del flusso locale.

E' stato sottolineato che la gestione delle ridondanze del sistema in oggetto risulta piuttosto complessa. Infatti, per quanto concerne la determinazione degli angoli di incidenza e derapata, è indispensabile disporre contemporaneamente dei segnali forniti da due sensori differenti. Le stime disponibili sono dunque sei: si tratta quindi di una gestione non-standard delle ridondanze ed in merito la letteratura fornisce scarsissime informazioni. Inoltre poiché è possibile il verificarsi di avarie sia dei sensori angolari che di pressione, occorre prevedere diverse modalità operative in funzione delle possibili combinazioni di guasto.

Al momento in letteratura vengono descritte, solamente, tecniche di gestione delle ridondanze relative a casi molto semplici. La metodologia sviluppata e

presentata nel lavoro può ritenersi dunque una risposta innovativa alle problematiche sopra esposte.

La procedura di elaborazione dati aria è stata interamente realizzata utilizzando *Stateflow*<sup>®</sup>, un *toolbox* innovativo di *Matlab*, con il quale è possibile simulare il comportamento di sistemi anche complessi. Questo *toolbox* si è rivelato uno strumento molto potente di progettazione grafica di sistemi basati sulla teoria delle macchine a stati finiti (*Finite State Machine*, FSM), che consente di rappresentare un sistema complesso che evolve nel tempo in base agli eventi che si verificano. L'elemento qualificante di questo *software* risiede proprio nella possibilità di gestire e visualizzare gli stati attivi durante una generica simulazione. Questo lavoro rappresenta dunque un esempio concreto di applicazione di *Stateflow*<sup>®</sup> alla realizzazione dei diversi modi operativi del sistema dati aria. E' emerso inoltre che, rispetto ad una programmazione tradizionale, *Stateflow*<sup>®</sup> riduce notevolmente sia i tempi di ottimizzazione che di calcolo, in quanto i modi non attivi non occupano risorse.

E' stato affrontato uno studio accurato dell'impatto delle possibili avarie dei sensori sull'accuratezza di stima dei parametri di volo, prendendo in considerazione, mediante un approccio alla Monte Carlo, le caratteristiche statistiche dei segnali forniti dalle sonde. I risultati ottenuti sono soddisfacenti in quanto anche in presenza di sonde in avaria, permangono ampie zone dell'involuppo di volo in cui, pur essendoci un degrado delle prestazioni, l'accuratezza rientra entro limiti accettabili.

Relativamente a questo argomento di ricerca, si pensa, in futuro di sviluppare nuove metodologie per la ricostruzione dei dati aria e per la gestione dei modi operativi del sistema, basati sull'utilizzo di tecniche di intelligenza artificiale.

Nella seconda parte del lavoro è stato affrontato lo sviluppo di un sensore innovativo, ispirato ai sistemi di tipo *Flush Air Data System* (FADS). La geometria ipotizzata è stata analizzata tramite CFD ed è stato definito un *database* preliminare di pressioni locali, su cui sono state allenate le reti neurali per la ricostruzione dei parametri di volo. Ciò si è rilevata una valida alternativa al calcolo basato su funzioni polinomiali approssimanti.

La metodologia utilizzata per definire i *set* di allenamento, validazione e *test* ha condotto a risultati soddisfacenti in termini di errore commesso sui parametri ricostruiti.

Nonostante il crescente interesse, anche industriale nei confronti di tali sistemi, la letteratura risulta ancora povera di lavori sull'argomento. Il presente dottorato ha gettato dunque le basi per una ricerca in tal senso: lo studio di reti dedicate alla ricostruzione della pressione, la definizione di architetture neurali con un numero maggiore di ingressi e capaci di rilevare le avarie oltre

---

ad un possibile ampliamento dell'involuppo di volo rappresentano gli futuri sviluppi di tale attività.



## Bibliografia

- [1] Haering, E. A., *Airdata Measurement and Calibration*, NASA Technical Memorandum 104316, Edwards, California, 1995.
- [2] Collins, R. P. G., *Fly-by-Wire Flight Control*, Computing and Control Engineering Journal, Agosto 1999.
- [3] Burns, B. R. A., *Pay-offs and Pitfalls of Fly-by-Wire*, Interavia Aerospace Review, Dicembre 1989.
- [4] *Fly-by-Wire Flight and Control Configured Vehicles-Reward and Risks*, The Aeronautical Journal, Febbraio 1975.
- [5] Favre, C., "Fly-by-Wire Flight for Commercial Aircraft: the Airbus Experience," *International Journal of Control*, Vol. 59, No. 1, Dicembre 1994.
- [6] Droste, C. S. and Walker, J. E., *The Genral Dynamics Case Study on the F-16 Fly-by-Wire Flight Control System*, AIAA Professional Study Series.
- [7] Mc Lean, D., *Aircraft Flight Control System*, The Aeronautical Journal, Marzo 1999.
- [8] Lawford, J., "Calibration of the Airdata System and Flow Direction Sensors," *Agard Flight Test Techniques Series*, Vol. 1, No. 300, Settembre 1983.
- [9] Gracey, W., *Summary and Methods of Measuring Angle of Attack on Aircraft*, NACA Technical Report 4351, 1958.
- [10] Menzies, M. A., *Integrated Air Data Sensors*, The Aeronautical Journal, Aprile 2001.

- [11] Cervia, F., Denti, E., Galatolo, R., Poggi, V., and Schettini, F., *Procedure e modelli per la simulazione dei dati aria*, Documenti del Dipartimento Ingegneria Aerospaziale DDIA-10, Università di Pisa, Maggio 2006.
- [12] Cervia, F., Denti, E., Galatolo, R., and Schettini, F., *Modellizzazione dei segnali generati da sonde coniche per la misura dei dati aria*, Documenti del Dipartimento Ingegneria Aerospaziale DDIA-24, Università di Pisa, Ottobre 2001.
- [13] Cervia, F., Denti, E., Galatolo, R., Rosignoli, N., and Schettini, F., *SW di simulazione real time dei sensori dati aria ed iniezione failure*, Documenti del Dipartimento Ingegneria Aerospaziale DDIA-11, Università di Pisa, Giugno 2004.
- [14] Cervia, F., Denti, E., Galatolo, R., and Schettini, F., *Modello di gestione delle Failure del sistema dati aria*, Documenti del Dipartimento Ingegneria Aerospaziale DDIA-13, Università di Pisa, Giugno 2003.
- [15] Schettini, F., *Studio delle problematiche di integrazione del sistema sensori nel Flight Control System di moderni velivoli ad elevate prestazioni*, Tesi di Dottorato, Università di Pisa, Luglio 2005.
- [16] Harel, D., "Using Statecharts for Hardware Description," *IEEE*, Novembre 1987.
- [17] Solano, M., *Sviluppo di una sonda multi pressione mediante analisi CFD e algoritmi basati su reti neurali per la ricostruzione dei parametri di volo*, Tesi di Laurea, Università di Pisa, A.A. 2007-2008.
- [18] inc., F., *Fluent 6.1 User's Guide*, Febbraio 2003.
- [19] Maggiani, G., *Sviluppo di metodologie basate su reti neurali per l'elaborazione dei dati aria dei velivoli*, Tesi di Laurea, Università di Pisa, A.A. 2007-2008.
- [20] Di Fusco, C., *Ricostruzione degli angoli di incidenza e derapata del velivolo mediante elaborazioni dei dati aria basate su reti neurali*, Tesi di Laurea, Università di Pisa, A.A. 2005-2006.
- [21] Orsucci, S., *Sviluppo ed ottimizzazione delle logiche di gestione delle ridondanze in un moderno Flight Control System FBW*, Tesi di Laurea, Università di Pisa, A.A. 2005-2006.

- 
- [22] *Air Data Measurement and Calibration*, NASA Technical memorandum 104316.

## Elenco delle Pubblicazioni

Di seguito riporto i lavori presentati ai congressi nazionali ed internazionali, i documenti interni del Dipartimento di Ingegneria Aerospaziale DDIA realizzati durante il mio dottorato ed infine il lavoro di tesi che ho seguito in qualità di relatrice.

- E. Denti, R. Galatolo, V. Poggi, F. Schettini, Sistema Dati Aria: analisi preliminari di Performance Assessment Air Data Module in Flow Fail Mode  $\alpha$ ,  $\beta$  nose boom, Documento del Dipartimento Ingegneria Aerospaziale DDIA-15, Università di Pisa
- E. Denti, R. Galatolo, V. Poggi, F. Schettini, Calcolo degli errori del sistema Dati Aria in Flow Fail Mode,  $\alpha$ ,  $\beta$  nose boom, Documento del Dipartimento Ingegneria Aerospaziale DDIA-17, Università di Pisa
- Partecipazione al Congresso Autonomy in UAVs Final Workshop, Novembre 2005, Monaco, Germania
- A. Calia, V. Poggi, F. Schettini, Air Data Failure Management in a Full-Authority Fly-By-Wire Control System, Lavoro presentato al Congresso Internazionale, Institute of Electrical and Electronics Engineers IEEE, Ottobre 2006, Monaco, Germania
- F. Cervia, E. Denti, R. Galatolo, V. Poggi, F. Schettini, Procedure e modelli per la simulazione dei dati - aria, Documento del Dipartimento Ingegneria Aerospaziale DDIA-10, Università di Pisa
- Air Data Module Performance (Flow Fail and By Pass Modes), Report 4kc-tg000029-i Issue E, Luglio 2006
- A. Calia, V. Poggi, F. Schettini, Air Data System Failure Management in a Full-Authority Fly-By-Wire Control System, Lavoro presentato

al Congresso Internazionale IEEE, Ottobre 2006, Technical University, Monaco - Germania

- A. Calia, R. Galatolo, V. Poggi, F. Schettini, Sviluppo di una sonda multi-pressione e di algoritmi di elaborazione per la ricostruzione dei parametri di volo, Lavoro presentato al XIX Congresso Nazionale AIDAA, 17-21 Settembre 2007, Forlì (FC)
- V. Poggi, Modellazione in ambiente Matlab/Stateflow di un sistema multi-stato per la ricostruzione dei parametri di volo, Lavoro presentato al XIX Congresso Nazionale AIDAA, 17-21 Settembre 2007, Forlì (FC)
- A. Calia, V. Poggi, F. Schettini, Angles of attack and sideslip reconstruction using neural networks, Lavoro presentato al 1th CEAS European Air and Space Conference, 10-13 September 2007, Berlino, Germania
- A. Calia, R. Galatolo, V. Poggi, F. Schettini, Multi-hole probe and elaboration algorithms for the reconstruction of the air data parameters, Lavoro presentato al Congresso Internazionale, Institute of Electrical and Electronics Engineers IEEE, International Symposium on Industrial Electronics ISIE, Giugno 2008, Anglia Ruskin University, Cambridge, UK
- M. Solano, Sviluppo di una sonda multi pressione mediante analisi CFD e algoritmi basati su reti neurali per la ricostruzione dei parametri di volo, Tesi di Laurea, Università di Pisa, A.A. 2007-2008,

Parallelamente all'attività di ricerca descritta, è stata svolta un'attività di supporto alla didattica relativa al corso di Dinamica del Volo con il Prof. E. Denti ed al corso di Dinamica del Volo dell'Elicottero con il Prof. A. Salvetti.

Corsi di approfondimento seguiti:

- Metodi numerici per le equazioni differenziali, Prof. F. Beux, Scuola Normale Superiore
- Introduzione all'uso di Mathematica, Prof. F. Romani, Dipartimento di Informatica
- Metodi di Ottimizzazione, Prof. M. Pappalardo, Dipartimento di Matematica Applicata
- Corso di Inglese scientifico, Prof. A. Wallwork

## Seminari:

- La Fluidodinamica computazionale in campo industriale, Ing. Mazzini, Ing. Moretti, Dipartimento Ingegneria Meccanica Nucleare e Produzione DIMNP San Piero a Grado, Pisa
- Collaborative control of unmanned vehicles, Prof. J. Karl Hedrick, Dip. Ing. Meccanica, Università della California, Berkeley
- The machine vision, Prof. M. Napolitano, West Virginia University WVU
- Implementable and Efficient trajectory design for an autonomous underwater vehicle, Prof. Ryan Smith, Dip. Ing. Meccanica, Hawaii
- Human machine vision with application to vehicle simulation and robot surgery, Prof. F. Cardullo, Dip. Ing. Meccanica, NY, USA

## Elenco delle figure

1.1	Schema di un sistema di controllo FBW. . . . .	2
1.2	Componenti fondamentali di un FCS/FBW . . . . .	4
1.3	Effetto scala di un CCV. . . . .	5
1.4	Confronto di prestazioni. . . . .	5
1.5	<i>Gust alleviation control</i> . . . . .	6
1.6	<i>Northrop-Grumman B2 Spirit</i> . . . . .	7
1.7	<i>Lockheed F117 Nighthawk</i> . . . . .	7
1.8	FCS/FBW con <i>backup</i> meccanico per l' <i>Airbus A320</i> . . . . .	10
1.9	Ridondanza di un FCS/FBW e delle sue parti. . . . .	11
1.10	Superfici di controllo di un velivolo intercettore leggero. . . . .	12
1.11	Distribuzione di pressione attorno ad un velivolo da trasporto. . . . .	17
1.12	Sensore <i>pivoted vane</i> . . . . .	18
1.13	Sensore a testa sferica. . . . .	18
1.14	Sonda <i>null-seeking pressure tube</i> . . . . .	19
1.15	Installazione tipo <i>nose boom</i> . . . . .	20
1.16	Installazione tipo <i>wingtip</i> . . . . .	20
1.17	Installazione in fusoliera. . . . .	20
2.1	Posizione in fusoliera e sistemi di riferimento delle sonde dati aria. . . . .	24
2.2	Vista tridimensionale della sonda dati aria. . . . .	24
2.3	Dati di ingresso ed uscite della procedura di elaborazione. . . . .	25
2.4	Schematizzazione degli effetti delle velocità angolari $Q_B$ e $R_B$ . . . . .	28
2.5	<i>Level 0</i> della procedura di simulazione dati aria. . . . .	30
2.6	Pannello di inserimento delle avarie sulla sonda 1. . . . .	31
2.7	Pannello di inserimento delle avarie sulla sonda 1. . . . .	32
2.8	<i>Level 0</i> del sistema dati aria. . . . .	33
2.9	Logica di funzionamento dell'algoritmo di <i>monitoring</i> angoli. . . . .	37

2.10	Algoritmo di <i>monitoring</i> sulle pressioni: individuazione di una sonda in avaria partendo da condizioni di pieno funzionamento.	38
2.11	Algoritmo di <i>monitoring</i> sulle pressioni: individuazione di una sonda in avaria partendo da un'avaria dichiarata. . . . .	38
2.12	Visualizzazione del passaggio dalla condizione di <i>Low Speed</i> a quella di <i>High Speed</i> . . . . .	41
2.13	Modi operativi della procedura di elaborazione dati aria. . . .	44
2.14	Modello Simulink delle modalità operative. . . . .	45
2.15	Esempio di <i>bubble diagram</i> . . . . .	47
2.16	Stati e transizioni in un diagramma <i>Stateflow</i> . . . . .	49
2.17	Diagramma <i>Stateflow</i> : la <i>connective junction</i> . . . . .	49
2.18	Diagramma <i>Stateflow</i> : attivazione <i>and/or</i> . . . . .	50
2.19	Diagramma <i>Stateflow</i> : visualizzazione dei risultati. . . . .	51
2.20	Procedura di elaborazione angoli. . . . .	52
2.21	Procedura di elaborazione angoli. . . . .	53
2.22	Procedura di elaborazione angoli. . . . .	54
2.23	Procedura di elaborazione pressioni. . . . .	55
2.24	Attivazione del voting sulle pressioni in presenza di due avarie.	56
2.25	Mappa dell'errore in $[\circ]$ commesso su $\alpha$ , a Mach 0.4, in condizione di piena operatività. . . . .	59
2.26	Mappa dell'errore in $[\circ]$ commesso su $\beta$ , a Mach 0.4, in condizione di piena operatività. . . . .	60
2.27	Mappa dell'errore in $[\circ]$ commesso su $\alpha$ , a Mach 0.4, in presenza di una sonda in avaria. . . . .	61
2.28	Mappa dell'errore in $[\circ]$ commesso su $\beta$ , a Mach 0.4, in presenza di una sonda in avaria. . . . .	62
3.1	Esempio di sonda multi funzione per il calcolo degli angoli di assetto (dimensioni in cm). . . . .	65
3.2	Esempio di sistema FADS installato sullo Shuttle X-33. . . .	65
3.3	Disposizione delle prese di pressione del sistema FADS sull'X33.	66
3.4	Velivolo non abitato dotato di apparecchiatura di monitoraggio del territorio. . . . .	66
3.5	Vista dall'alto del corpo principale della sonda di pressione. .	67
3.6	Vista tridimensionale della sonda completa. . . . .	68
3.7	Flusso logico per la realizzazione della <i>mesh</i> e dell'analisi CFD.	69
3.8	Modello semplificato della sonda. . . . .	70
3.9	Dimensioni dei nove <i>box</i> realizzati intorno al modello. . . . .	71
3.10	Vista tridimensionale dei <i>box</i> . . . . .	71
3.11	Applicabilità degli elementi di superficie ai relativi modelli di riempimento. . . . .	72



3.12	Modelli di riempimento. . . . .	72
3.13	Nodi posizionati sulla linea di contorno tra due superfici. . . .	73
3.14	Applicabilità degli elementi di volume ai relativi modelli di riempimento. . . . .	74
3.15	Variazione della dimensione degli elementi di volume in pros- simità della calotta sferica. . . . .	74
3.16	Visualizzazione del flusso logico per effettuare le simulazioni CFD. . . . .	77
3.17	Posizione delle prese di misura sugli archi principali della sonda.	79
3.18	Flusso logico per verificare la convergenza della soluzione. . .	79
3.19	Andamento dei residui in funzione del numero delle iterazioni.	80
3.20	Visualizzazione dei vettori velocità sul piano $xy$ per una simu- lazione ad $\alpha$ , $\beta$ nulli e $v = 70m/s$ . . . . .	81
3.21	Andamento della $\Delta P$ in funzione del punto di controllo, al variare di $\alpha$ per $\beta = 0^\circ$ e $v = 20m/s$ . . . . .	82
3.22	Andamento della $\Delta P$ in funzione del punto di controllo, al variare di $\alpha$ per $\beta = 0^\circ$ e $v = 70m/s$ . . . . .	82
3.23	Andamento della $\Delta P$ in funzione della velocità per $\alpha$ e $\beta$ nulli.	83
3.24	Andamento della $\Delta P$ in funzione della velocità per $\alpha = 10^\circ$ e $\beta = 0^\circ$ . . . . .	83
3.25	Visualizzazione delle simmetrie geometriche: andamento della $\Delta P$ in funzione del punto di controllo per $v = 20m/s$ . . . . .	84
3.26	Griglia delle simulazioni effettuate a $v = 20m/s$ in funzione di $\alpha$ e $\beta$ misurati in gradi. . . . .	84
3.27	Griglia delle simulazioni effettuate a $v = 70m/s$ in funzione di $\alpha$ e $\beta$ misurati in gradi. . . . .	85
3.28	Andamento dell'errore commesso durante le interpolazioni in funzione del punto di controllo. . . . .	86
3.29	Visualizzazione tridimensionale della $\Delta P$ relativa al punto di controllo centrale, in funzione di $\alpha$ e $\beta$ , per $v = 20m/s$ . . . . .	86
3.30	Neurone biologico: visualizzazione dei costituenti. . . . .	88
3.31	Eventi significativi per lo sviluppo di tecniche neurali. . . . .	89
3.32	Neurone artificiale: caratteristiche e costituenti. . . . .	91
3.33	Esempio di rete completamente connessa. . . . .	92
3.34	Reti stratificate. . . . .	93
3.35	Fasi dell'algoritmo di <i>Back-Propagation</i> . . . . .	95
3.36	Visualizzazione dell'insieme di allenamento, validazione e test.	96
3.37	Andamento della pressione in funzione di $\beta$ e delle prese di misura. . . . .	97

3.38	Andamento della pressione in funzione di $\alpha$ e delle prese di misura. . . . .	98
3.39	Punti scelti per posizionare i sensori (ingressi ottimali per le reti neurali). . . . .	99
3.40	Andamento della pressione $\Delta P$ in funzione di $\alpha$ nei punti scelti come sensori di misura. . . . .	100
3.41	Andamento della pressione $\Delta P$ in funzione di $\beta$ nei punti scelti come sensori di misura. . . . .	100
3.42	Esempio di denominazione di una generica rete. . . . .	101
3.43	Schema di calcolo dell'errore commesso dalle reti nella ricostruzione dei parametri di volo. . . . .	101
3.44	Tabella degli errori (in $^{\circ}$ ) commessi dalle reti nella ricostruzione di $\alpha$ . . . . .	107
3.45	Distribuzione, nell'involuppo $(\alpha, \beta, v)$ , degli errori (in $^{\circ}$ ) commessi dalla rete su $\alpha$ . . . . .	108
3.46	Tabella delle eccedenze delle reti che ricostruiscono $\alpha$ . . . . .	108
3.47	Tabella degli errori (in $^{\circ}$ ) commessi dalle reti nella ricostruzione di $\beta$ . . . . .	109
3.48	Distribuzione, nell'involuppo $(\alpha, \beta, v)$ , degli errori (in $^{\circ}$ ) commessi dalle reti nella ricostruzione di $\beta$ . . . . .	110
3.49	Tabella delle eccedenze delle reti che ricostruiscono $\beta$ . . . . .	110
3.50	Tabella degli errori (in [m/s]) commessi dalle reti nella ricostruzione di $v$ . . . . .	111
3.51	Distribuzione, nell'involuppo $(\alpha, \beta, v)$ , degli errori (in $^{\circ}$ ) commessi dalle reti nella ricostruzione di $v$ . . . . .	112
3.52	Tabella delle eccedenze delle reti che ricostruiscono $v$ . . . . .	112
3.53	Tabella degli errori (in [m/s]) commessi dalle reti nella ricostruzione di $v$ sul <i>training set</i> ampliato. . . . .	113
3.54	Tabella degli errori (in [m/s]) commessi dalle reti nella ricostruzione di $\alpha$ e $\beta$ . . . . .	114
3.55	Distribuzione, nell'involuppo $(\alpha, \beta, v)$ , degli errori su $\alpha$ (in $^{\circ}$ ), commessi dalle reti che ricostruiscono $\alpha$ e $\beta$ . . . . .	115
3.56	Distribuzione, nell'involuppo $(\alpha, \beta, v)$ , degli errori su $\beta$ (in $^{\circ}$ ), commessi dalle reti che ricostruiscono $\alpha$ e $\beta$ . . . . .	115
3.57	Andamento dell'errore MSE durante l'allenamento della rete sulla velocità, con ingressi normalizzati, provenienti dalle prese (0 5 19 33 47). . . . .	116
3.58	Andamento dell'errore SSE durante l'allenamento della rete sulla velocità, con ingressi normalizzati, provenienti dalle prese (0 5 19 33 47). . . . .	116

## Elenco delle tabelle

2.1	Coppie disponibili al variare delle sonde in avaria. . . . .	35
2.2	Valori degli angoli costituenti i gruppi. . . . .	36
2.3	Algoritmi di <i>voting</i> angoli. . . . .	42
2.4	Algoritmi di <i>voting</i> pressioni. . . . .	42

## Elenco dei simboli

**FCS** *Flight Control System*

**FBW** *Fly-By-Wire*

**FCS** *Automatic Flight Control System*

**FCC** *Flight Control ComputerFly-By-Wire*

**CCV** *Control Configured Vehicle*

**PIO** *Pilot Induced Oscillations*

**DIA** *Dipartimento di Ingegneria Aerospaziale*

**FSM** *Finite State Machine*

**ASI** *Avionics Specialities Inc.*

**IMFP** *Integrated Multi Functional Probe*

**RVDT** *Rotary Variable Differential Transducer*

**AoA** *Angle of Attack*

**AoS** *Angle of Sideslip*

**CAS** *Calibrated Air Speed*

**ND** *Nose Drop*

**LG** *Landing Gear*

**ADS** *Air Data Simulation*

**FIP** *Failure Injection Panel*

**BIT** *Built In Test*

**STD** *State Transition Diagram*

**ABS** *Air Break System*

**GUI** *Graphical User Interface*

**FADS** *Flush Air Data System*

**LE-FADS** *Leading Edge FADS*

**CFD** *Computational Fluid Dynamics*

**HARV** *High Research Vehicle*

**DNS** *Direct Numerical Simulation*

**LES** *Large Eddy Simulation*

**RANS** *Reynolds Averaged Navier-Stokes equation*

**NN** *Neural Network*

**MLP** *Multi Layer Perception*

**ES** *Early Stopping*

**BR** *Bayesian Regularization*